Journal of Soft Computing in Civil Engineering 9-1 (2025) 122-157

Contents lists available at SCCE

Journal of Soft Computing in Civil Engineering

Journal homepage: www.jsoftcivil.com

A Comparative Study between Different Machine Learning Algorithms for Estimating the Vehicular Delay at Signalized Intersections

Yazan Ibrahim Alatoom [1,*]; Abdallah B. Al-Hamdan [1]

1. Ph.D. Student, Department of Civil, Construction, and Environmental Engineering, Iowa State University, Ames, Iowa, 50011, United States

* Corresponding author: *yalatoom@iastate.edu*

https://doi.org/10.22115/scce.2024.418800.1725

ARTICLE INFO

Article history:
Received: 30 September 2023
Revised: 03 April 2024
Accepted: 14 April 2024

Keywords:
Vehicle delay estimation;
Traffic signal delay modeling;
Machine learning for delay prediction;
Signalized intersection delay;
Stop delay models;
Data-driven delay models;
Comparative study of delay algorithms;
Random forest for delay estimation.

ABSTRACT

The delay at signalized intersections is a crucial parameter that determines the performance and level of service (LOS). The estimation models are commonly used to model delay; however, inaccurate predictions from these models can pose a significant limitation. Consequently, this study aimed to compare a wide array of machine learning algorithms, including Artificial Neural Networks (ANN), Random Forest (RF), decision tree, Support Vector Machine (SVM), K-Nearest Neighbor (K-NN), AdaBoost, Gradient Boost, XGBoost, and Partial Least Squares (PLS) regression. A comprehensive evaluation was undertaken across prediction accuracy, training-testing performance discrepancy, sensitivity to outliers, computational time cost, and model robustness. Additionally, the proposed methods were benchmarked against the Highway Capacity Manual (HCM), Webster, and Akçelik models. The results demonstrated that the RF model exhibited the most balanced performance across the specified criteria, with an average error below 4% and a rating of 35 out of 45 according to the proposed criteria. Moreover, the findings revealed that adopted analytical models should not be employed for vehicular delay estimation without calibration, as RMSE values were about 5 to 58 times higher than other models, varying by model.

How to cite this article: Alatoom YI, Al-Hamdan AB. A comparative study between different machine learning algorithms for estimating the vehicular delay at signalized intersections. J Soft Comput Civ Eng 2025;9(1):122–157. https://doi.org/10.22115/scce.2024.418800.1725

2588-2872/ © 2025 The Authors. Published by Pouyan Press.

This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

The delay at signalized intersections significantly impacts traffic operations, increasing travel time and reducing quality of service. Poor traffic planning is a common issue that leads to unsatisfactory service levels at signalized intersections. Traffic delay at signalized intersections refers to the extra time vehicles spend traversing the intersection [1,2]. Three types of delay exist at signalized intersections: control delay, stopped delay, and approach delay [1,3]. Control delay represents the vehicular delay at the signalized intersection. The stopped delay represents the time vehicles still stopped at the intersection until the next green phase begins. Approach delay includes the stopped delay with adding the time loss due to vehicle deceleration and re-acceleration at the signalized intersection [1].

Traditionally, two main methods are used to evaluate traffic delays at signalized intersections. The first involves field studies, an approach that proves costly and labor-intensive, particularly in developing countries such as Jordan. The second employs predictive models, which have been widely adopted by transportation agencies to mitigate the challenges of field studies. However, the significant shortcoming of such models lies in their inaccurate predictions. Precise predictions are imperative for efficacious traffic planning, as inaccurate forecasts may lead in inaccurate evaluation and planning. [4].

Numerous studies have been undertaken with the aim of developing traffic delay models for signalized intersections. The initial endeavor in this area was conducted by Webster [5]. A mathematical model was developed for homogenous traffic based on the concept of the degree of saturation at the signalized intersection. Federal Highway Administration (FHWA) introduced the Highway Capacity Manual (HCM) model [6,7], which is the most common model for estimating signalized delay based on several traffic parameters. Akcelik [8] developed a generalized model based on the HCM model, aiming to introduce an international model to estimate traffic delay at signalized intersections. The study also compared the HCM [9] model with Akcelik [10] and Richardson et al. [11] models. The analytical models were primarily based on homogeneous traffic conditions and theoretical assumptions that may not adequately capture the complexities of real-world signalized intersections, especially in the presence of heterogeneous traffic.

The statistical approach has been widely adopted by researchers to develop delay estimation models based on a range of traffic, geometric, and signal timing parameters, including traffic volume, volume-to-capacity ratio, effective green time, and lane width. Thamizh Arasan and Jagadeesh [12] used a probabilistic approach to estimate signalized delay caused by traffic based on traffic flow volume, saturation flow rate, cycle timing, and the degree of saturation of the evaluated signalized intersections. Ahmed et al. [13] developed analytical macroscopic models to evaluate the delay at signalized intersections for extended control periods, taking into account downstream time-variable congestion as influenced by green ratios and geometric combinations. The models were tested in a closed system with varying traffic and geometry parameters. The result showed that delay caused by upstream intersection varies with time and reaches constant if

downstream flow and signal measures stabilize and start replicating over time. Korkmaz and Akgüngör [14] developed three types of delay estimation models: linear, exponential, and quadratic. The models were a function of two parameters: green ratio and degree of saturation. The result showed that the developed model was found to be better in delay estimation than HCM [15] and Akcelik [10] models with high R^2 and low Mean Square Error (MSE). Sofia et al. [16] developed an empirical regression model to estimate the delay at signalized intersections in Iraq. The independent variables were based on traffic parameters and signalized intersection geometry. The adjusted R^2 reached 0.995. Although the R^2 was very high, the number of data points was small and insufficient to develop a generalized delay model. While these models provide a systematic framework for delay estimation, they often rely on simplified assumptions and theoretical constructs that may not fully capture the inherent complexities and dynamic nature of signalized intersections, especially under heterogeneous traffic conditions.

Due to the rapid advancement of technologies in recent decades, utilizing modern techniques to solve novel problems, rather than relying solely on traditional approaches, has become imperative. One example of embracing new technologies is the use of Artificial Intelligence (AI). Several recent studies have explored the application of machine learning techniques in various civil engineering domains. These include the evaluation of machine learning algorithms for seismic retrofit cost estimation [17], the development of a hybrid random-forest-based subset simulation method for reliability analysis of marine structure piles [18], the use of artificial intelligence models to predict the axial compressive strength of recycled aggregate concrete-filled circular steel tube columns [19], the application of a hybrid MARS-GOA model for reliability assessment of compressive and splitting tensile strength prediction of roller compacted concrete pavement [20], estimating pier scour depth using surrogate models and comparing them with empirical formulations [21], and an evolutionary neuro-fuzzy-based approach to estimate the compressive strength of eco-friendly concrete containing recycled construction wastes [22].

AI and non-parametric approaches have been employed in several areas of transportation engineering research, including traffic safety research [23], traffic planning [24], traffic congestion [25], and many other applications. New technologies are used to estimate traffic delay at signalized intersections. Qiao et al. [26] used fuzzy-based logic to develop a signalized intersection delay model. The results showed that the proposed model was useful for delay estimation at signalized intersections and produced lower errors than other conventional models. Artificial Neural Network (ANN) was also used to develop delay prediction models. Sazi Murat and Baskan [27] developed an ANN model based on signal timing and traffic influence measures. They compared the developed model with the HCM [15] and Webster [5] models. The result revealed that the ANN model was more accurate than other models. Preethi et al. [28] developed an ANN model for predicting the delay at the signalized intersections under heterogeneous traffic conditions. The inputted variables were based on traffic volume parameters and signal timing. The results reported that the developed model was better than the HCM [1] and Webster [5] models in terms of prediction accuracy. Dogan et al. [29] studied the potential of using ANN to predict the stopped delay and number of stopped vehicles at signalized intersections generated from simulation software. Intersection approach volumes, cycle length,

and left-turn lane existence were examined as input variables of the model. The result showed a very good estimation of the developed ANN model.

Neuro-fuzzy systems were also examined in many types of research to estimate the vehicular delay at signalized intersections. This approach benefits from both the linguistic, human-like reasoning of fuzzy systems and the efficient computational capabilities of neural networks [30–33]. Hasiloglu et al. [34] compared the regression approach, Single Simulation Model (SSM), Webster [5] model, HCM [1] model, and Adaptive Neuro-Fuzzy Inference System (ANFIS) in terms of prediction performance for the delay in the signalized intersections. The results showed that ANSFIS, SSM, and regression models were found to be the closest to the delay observed values. A similar study was conducted by Gökdağ et al. [35], which developed an ANFIS model for signalized intersections delay and was compared with Doherty Delay Formula (DDF), Webster [5], and HCM [6] models. The results showed that the ANFIS approach could be used for delay estimation at signalized intersections. Sazi Murat [36] developed ANN and ANFIS models for delay estimation at signalized intersections and compared these models with HCM [15], Webster [5], and Akcelik [10] models. The testing results reported that the average error of ANN and ANFIS models was found to be less than other models.

A study by Gilani et al. [37] investigated the performance of bus special lanes using statistical analysis and optimization of signalized intersection delay through machine learning methods. The results showed that the ANFIS model best predicted and optimized the intersection delay along the bus lane. Mahmoud et al. [38] focused on predicting cycle-level traffic movements at signalized intersections using machine learning models. They found that the Gated Recurrent Unit (GRU) model using data from six previous cycles outperformed Long Short-Term Memory (LSTM) and XGBoost in forecasting through and left-turn movements. Another study by Ranpura et al. [39] estimated vehicle control delay at a signalized intersection under heterogeneous traffic using AI techniques, finding that the random forest model outperformed other machine learning and conventional delay estimation methods.

Bagdatli [40] used the Gene-Expression Programming (GEP) technique to develop three delay models for signalized intersections using different inputted variables (cycle time, red time, traffic volume). It was found that the GEP technique predicted the delay accurately. Bagdatli and Dokuz [41] examined Random Forest, K-NN, SVM, and XGBoost algorithms for estimating the control delays at signalized intersections in Turki. One limitation of this prior work was the utilization of queue length as a variable, which exhibits a high correlation with delay and necessitates equivalent field measurement effort. Despite this study demonstrating the accuracy of machine learning models for delay estimation, it did not examine other significant factors, such as the time cost of each model. Moreover, this and other previous studies focused on prediction performance in terms of the deviation of the predicted values from the actual values without accounting for variation in traffic data. The current study thus endeavors to address these gaps in the literature by introducing novel metrics and tests, in conjunction with the prediction performance measures presented in recent works, to comprehensively evaluate and compare a wide range of machine learning algorithms.

2. Research significance

The utilization of machine learning algorithms in traffic engineering has experienced significant growth in recent decades, owing to their robust approximation capabilities in modeling complex data, which often surpass the performance of analytical and conventional methods. Previous studies employing machine learning for vehicular delay estimation at signalized intersections have consistently indicated that these techniques hold greater promise than conventional models. However, a review of the literature reveals several limitations:

1- Most of these studies were conducted under saturation conditions and didn't take into account the over-saturation condition (volume to capacity > 1) and with small ranges of delay data.

2- Also, most of these studies just focused on one or two machine learning algorithms and were compared with conventional models, such as the HCM model [1] and Webster [5], without taking into account other methods, such as empirical regression with the proper transformation of independent and dependent variables or other machine learning methods.

3- Furthermore, the comparisons of models in these studies just focused on the performance of the proposed methods in terms of prediction accuracy without considering other important factors that could affect the adoption of these models by the transportation agencies, such as time efficiency or other factors.

Accordingly, the primary objective of this study was to provide a comprehensive comparison between a wide range of machine learning models for estimating stopped delay at signalized intersections. The comparison evaluated prediction accuracy, divergence in performance between training and testing sets, sensitivity to outliers, computational time costs, and model robustness against data variation. Additionally, the study benchmarked these techniques against prevalent analytical methods documented in the literature, including the HCM, Webster, and Akçelik models.

3. Background on machine learning models

3.1. Artificial neural networks

ANN represents a type of machine learning technique characterized as non-parametric, as they have the capacity to learn from data and estimate complex functions. [42–44]. ANN was adopted in different transportation engineering applications such as planning, traffic operation, and control, as well as transportation design applications [45]. ANN was also used widely in transport forecasting [46]. It is also used in many fields, such as face recognition and object detection [47].

The structure of ANN mainly consists of three components called layers: the input, the hidden, and the output layers. The input layer represents the inputted parameters, the hidden layer is responsible for linking the input layer with the output layer and could be one layer or more

depending on the problem being solved, and the output layer is the final product of the last layers [42,48,49].

Each layer in the network structure contains nodes called artificial neurons. Each neuron in the layers interacts with each other in the next and previous layers by weights, summing function, bias, and activation function [42,48]. The following shows the general formula for the ANN model:

$$D_s = \alpha_0 + \sum_{j=1}^{n}\left(\alpha_j \text{tansig}\left\{\sum_{i=1}^{m} w_{ij}x_i + \beta_{0j}\right\}\right) \quad (1)$$

Where, D_s is the output of the ANN model; x_i is the input parameters, m is the number of input neurons, n is the number of hidden layers; w_{ij} is the weights between input and hidden layers; α_j is the weight between the previous and next layer; α_0 and β_{0j} are the biases (constants).

In any ANN model, the data should be split into two sets: the training set and the testing set. These group sets are used in two stages: training and testing, respectively. In the training stage, the model learns from the data using the learning algorithm [42,48]. The testing stage is usually used to test the performance of the developed model [42,48].

3.2. Tree decision and random forest

A decision tree is a machine learning method that partitions the feature space into rectangular regions, iteratively splitting it based on the values of input features. It predicts the mean or mode of the target variable for the observations within each region. In regression problems, the predicted value is typically the mean of the target variable for observations within the partitioned area. The division of the feature space is determined by minimizing a loss function, such as the error term, to optimize predictive accuracy [50,51].

Random Forest (RF) is a non-parametric method that splits the data samples into classes or sub-samples and builds multiple decision trees to estimate the final result of the model [52–54]. RF operates by creating a multitude of decision trees (B) during training, where each tree outputs a prediction. The final prediction is then determined by aggregating the predictions of all individual decision trees [55,56]. This machine learning method showed a high ability to handle many complex relationships between input and output variables [51]. The RF is based on the theoretical concept of Classification and Regression Tree (CART), which splits the data into a set of regions and averages the observations within each region. The splitting points and variables determined by minimizing the loss function [50,51,57]. The method of CART suffered from the overfitting problem; therefore, RF is considered an improvement of the CART method [52].

The RF consists of two subsampling categories: row-subsampling and column-subsampling. The row-subsampling is a bootstrap number of versions of the training set, while column-subsampling takes random sets from the predictor variables [58]. These two subsampling types are very important in RF to avoid the overfitting problem. The major components of RF are the number of trees (B), the number of predictors on each tree (P), and the number of samples used in each tree (N) [59]. Fig. 1 shows an example of the structure of Tree decision and Random Forest.

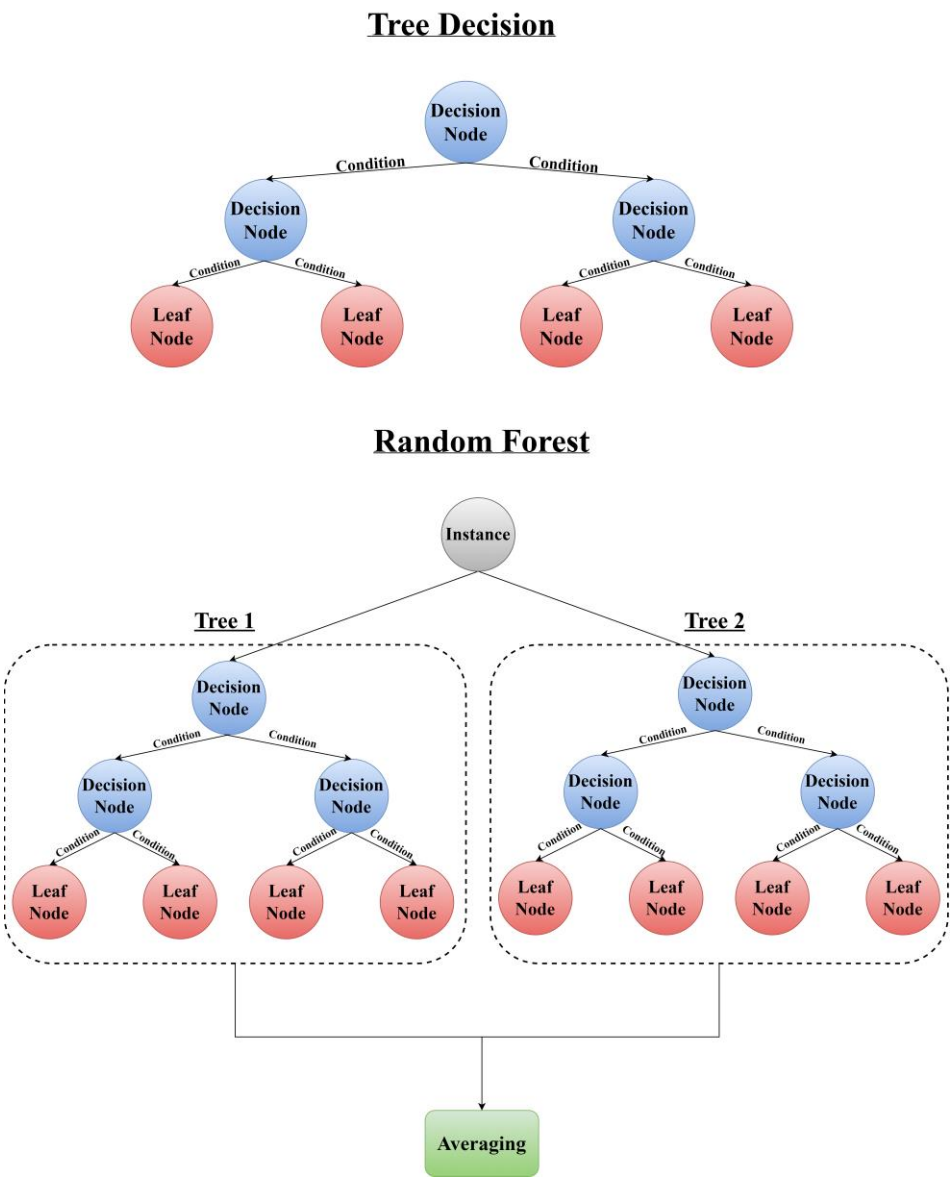

Fig. 1. The difference between Tree decision and Random Forest algorithms.

3.3. AdaBoost, gradient boost, and XGBoost

Adaptive Boost (AdaBoost) is a machine learning algorithm that enhances the performance of other models by combining weak learners, typically decision trees, through a weighted sum to produce a final prediction. [60]. On the other hand, Gradient Boost uses the same path as AdaBoost. However, it identifies the hard observation during the training phase rather than combining the weighted sum of observations. These models are usually used with tree decision, which is considered weak learners, to improve the performance of the training at each tree stage and then improve the final output accuracy. Fig. 2 and Fig. 3 show the methodology of AdaBoost and Gradient Boost models. As shown in Fig. 2, AdaBoost is based on a stump tree, which is the shortest tree decision. However, as shown in Fig. 3, Gradient boost first averages the target data and measures the error; then, the next tree decision is developed based on the previous error from other trees.

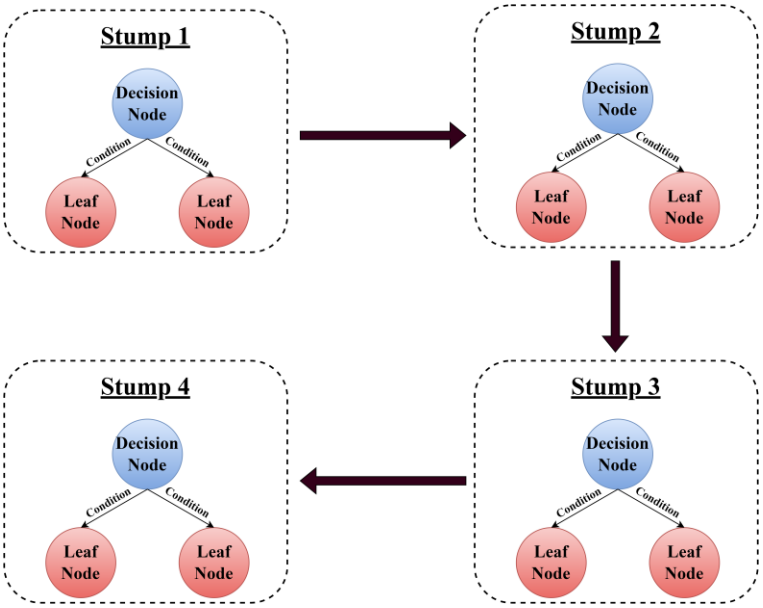

Fig. 2. The algorithm of the AdaBoost model.

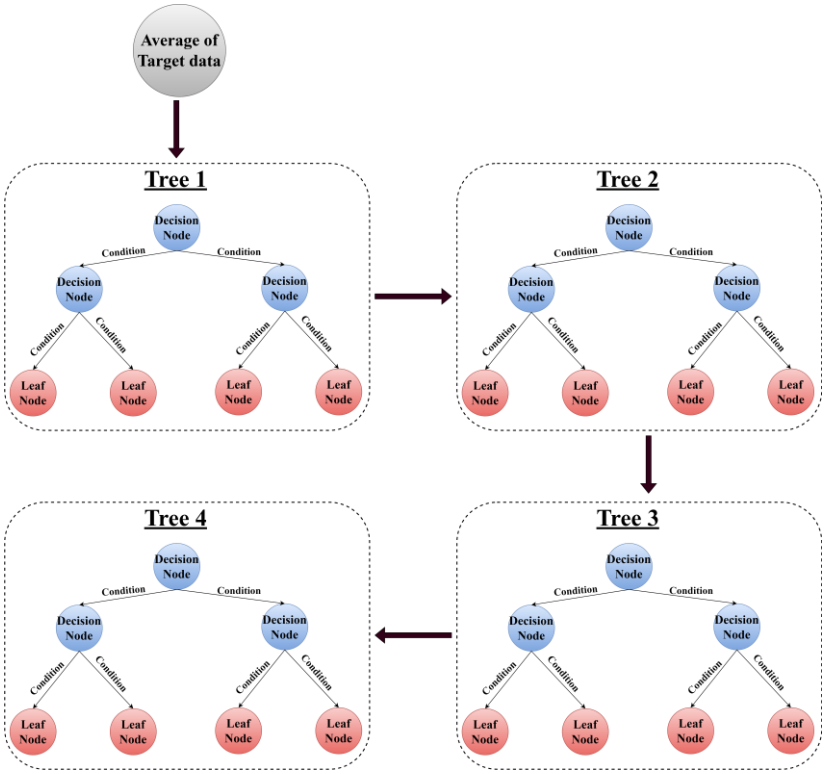

Fig. 3. The algorithm of the Gradient Boosting model.

The extreme gradient boosting (XGBoost) algorithm is a new version of Gradient Boost based on the conventional tree approach [41,61]. To decrease residual error and improve speed, the XGBoost algorithm creates new trees from previous ones [41,61]. Rather than employing the loss function's first derivative, the XGBoost method approximates it using second-order Taylor expansion [41]. Because of its structure, the XGBoost method is a fast and scalable technique; the primary purpose is to reduce the objective function by improving decision trees [41]. For

regression, the XGBoost method is used, and it is chosen over other incremental algorithms since it is faster and more efficient [41].

3.4. Support vector machine

Support Vector Machine (SVM) is a machine learning algorithm rooted in statistical theories, initially introduced by by Boser et al. [62] for classification purposes. SVM is based on the concept of SRMP (structural risk minimization principle) that addresses the problem of overfitting [63]. Traditional machine learning techniques, such as neural networks, are based on Empirical Risk Minimization (ERM) [64]. SVM was adopted in many fields of machine learning and was considered a very effective technique for classification and regression purposes [64–66].

The simplest type of SVM is based on the concept of the maximum margin classifier, which entails constructing a linear hyperplane to effectively classify and separate binary data by maximizing the margin between different classes. Fig. 4 shows the basic components of the SVM model. Because real-world data have nonlinear characteristics and cannot be classified using a simple linear hyperplane, kernel mappings were introduced to be an efficient method to project data into a higher dimensional feature space [67]. Kernel functions are used to map and transform inputted data into SVM and could be linear, polynomial, sigmoid, Radial Basis Function (RBF), and spline, each offering different ways to represent the data for optimal classification..

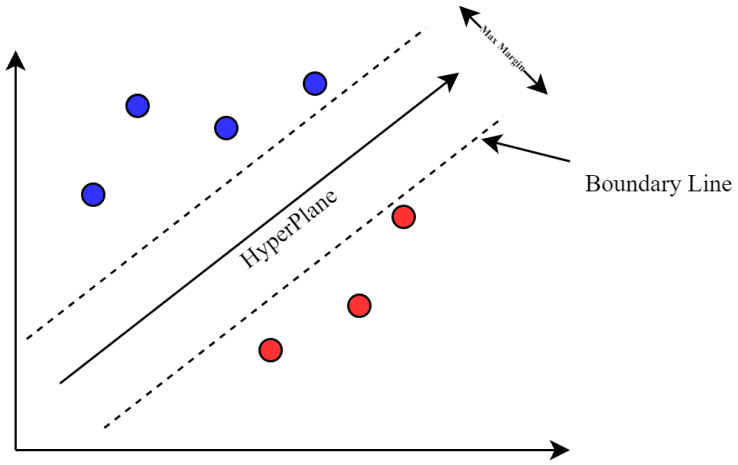

Fig. 4. The basic concepts of the SVM model.

To apply Support Vector Machines (SVMs) for regression tasks, a variant known as Support Vector Regression (SVR) could be utilized. Unlike traditional Ordinary Least Squares (OLS) regression, which minimizes the error term, Support Vector Regression (SVR) aims to minimize the weight vector (w) while handling errors through constraints (maximum error). The minimization equation for SVR is as follows [67,68]:

$$min\frac{1}{2}w^{T}w + C\sum_{i=1}^{N}(\delta_i) \quad (2)$$

Where, w is a vector that determines the orientation of the hyperplane; δ_i is a deviation from the margin; C is a hyperparameter.

3.5. K-Nearest neighbors

K-Nearest Neighbors (K-NN) is a type of machine learning algorithm first developed by Fix and Hodges [69] that belongs to the family of instance-based, lazy learning algorithms. It operates on the principle of similarity, where it searches for the K nearest data points in the dataset based on a chosen distance metric. Once the nearest neighbors are identified, the algorithm estimates the output for a new data point by aggregating the outputs of its nearest neighbors, typically through methods like taking the mode, average, or median of the output [70,71]. The choice of distance function is crucial in K-NN, as it determines how the algorithm measures the similarity between data points. Common distance functions include Euclidean distance, Manhattan distance, Mahalanobis distance, and Minkowski distance. Euclidean distance is perhaps the most widely used and intuitive distance metric, representing the straight-line distance between two points in the feature space. Manhattan distance, also known as city block distance, measures the distance between two points by summing the absolute differences of their coordinates. Mahalanobis distance, on the other hand, takes into account the distribution of the data and calculates the distance between a point and a distribution, considering the covariance matrix. The formulas for Euclidean and Mahalanobis distances are shown as follows:

$$M = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})} \tag{3}$$

$$\mathrm{E} = \sqrt{(\mathrm{x}_1 - \mathrm{y}_1)^2 + (\mathrm{x}_2 - \mathrm{y}_2)^2} \tag{4}$$

Where, M = Mahalanobis distance; E = Euclidean distance; $\vec{x}$ = vector of observations; $\vec{\mu}$ = Mean of observations; S = covariance matrix; x_1 and x_2 = observations x-axis coordinate; y_1 and y_2 = observations y-axis coordinate.

The algorithm of K-NN for regression begins by specifying a value for K. Then, it iterates through the training data, calculating the distance between the current point and each point in the dataset. Next, it selects the K-nearest labels based on their distances (shortest distances) and outputs the mean of these K-labels as the predicted value for the current point. [70]. Fig. 5 shows an example of the concepts of the K-NN model using different K values.

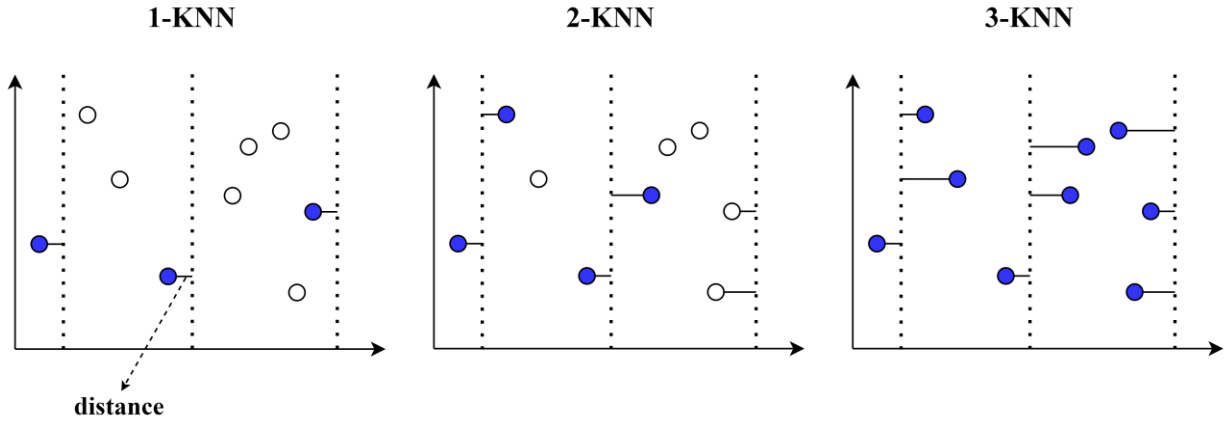

Fig. 5. The effect of K values on the K-NN model.

4. Data collection and description

The study utilized a dataset [72] collected from eight isolated signalized intersections in Amman, the capital of Jordan. The intersections included three and four-leg configurations with varying traffic conditions and geometries. The dataset comprises a total of 322 stopped delay data points per signal cycle, collected at 15–45-minute intervals. It consists of 50% original data and 50% augmented data generated from the original data to enhance the robustness of the models. The data were collected mostly on weekdays, excluding peak hours, to ensure representativeness and comprehensiveness. The isolated signalized intersections were preferred in this study to evaluate the delay caused by the intersection environment without being affected by other close-range intersections.

The data were processed as a function of four input variables: equivalent arrival rate (V) per hour, effective green time to cycle length ratio (g/C), degree of saturation (X_s), and lane group approach width (W). The arrival rate is not an instantaneous variable and must be aggregated for a time period. Therefore, it is collected for the equivalent one-hour interval. These variables were selected based on their higher Pearson correlation with the output variable (stopped delay) and their significance in modeling delay, as reported in the literature.

The data collection process involved both direct field data collection and a video camera positioned at an angle that could clearly capture the desired intersection. The average stopped delay, defined as the time vehicles stop until the light turns green, was measured according to the HCM model procedure [1], as described in Chapter 31. Each intersection typically has different phases and multiple lanes, so the data were collected and processed separately for each lane group. The concept of lane groups for each approach and the methodology employed for collecting delay and other variables on a per-lane-group basis are elucidated in Fig. 6.

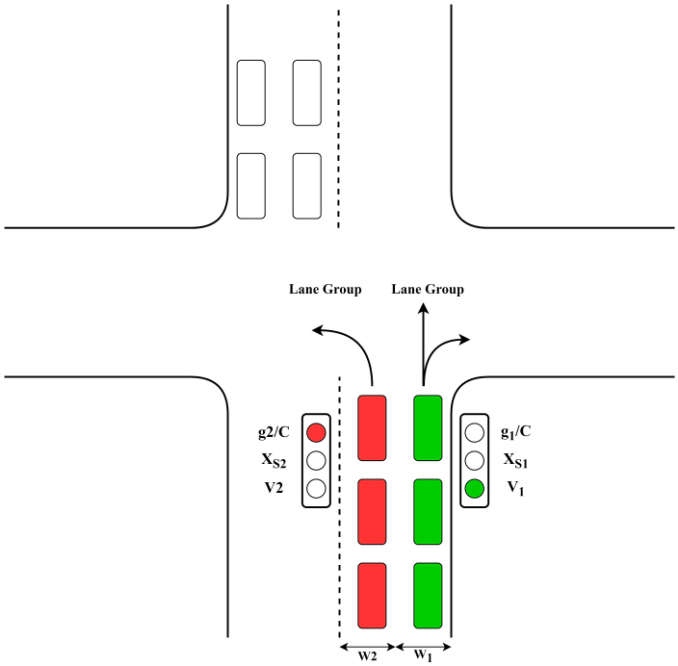

Fig. 6. An example of how the delay and other variables were collected.

Table 1 summarizes the statistical characteristics of the collected data, including the ranges of the independent and dependent variables. The dataset encompasses a wide range of traffic volumes, signal timings, stopped delays, and degrees of saturation, ensuring its suitability for training and testing machine learning models. The stopped delay (D_s) exhibits a substantial range from 4.57 to 254 s/veh, with a mean of 40.48 s/veh. The independent variables capture a diverse array of traffic conditions and intersection geometries, with g/C ranging from 0.05 to 0.58 (mean = 0.29), X_s from 0.01 to 1.76 (mean = 0.39), arrival rates (V) from 23 to 1252 veh/h (mean = 456.80 veh/h), and lane group width (W) from 3 to 14.72 meters (mean = 8.81 m). The dataset encompasses 322 data points, encompassing the stopped delay (s/veh) as the dependent variable. The data size is sufficient for training and testing according to the 1 in 10 rule of thumb [73]. According to this rule, at least 10 sample data were required for each predictor. In our study, there are 322 data points and four predictors, and the ratio is about 80, which is greater than 10, the recommended threshold.

Table 1
The statistical characteristics of independent and dependent variables.

Variable	Unit	Number	Minimum	Maximum	Mean	Standard Deviation
$\boldsymbol{D_s}$	s/veh	322	4.57	254	40.48	34.02
$\boldsymbol{g/C}$	-	322	0.05	0.58	0.29	0.15
$\boldsymbol{X_s}$	-	322	0.01	1.76	0.39	0.35
$\boldsymbol{V}$	veh/h	322	23.00	1252.00	456.80	255.30
$\boldsymbol{W}$	meter	322	3.00	14.72	8.81	2.50

The saturation flow rate in this study was measured according to the HCM model procedure [1], while the base saturation flow rate was assumed to be equal to 2050 pc/h/ln according to a study conducted in Jordan and Kuwait that recommend values for base saturation flow rate in the developing countries [74]. The following formula was used to measure the saturation flow rate for each lane group (HCM 2010):

$$S = (S_O)(f_w)(f_{HVg})(f_p)(f_{bb})(f_a)(f_{Lu})(f_{LT})(f_{RT})(f_{Lpb})(f_{Rpb})(f_{wz})(f_{ms})(f_{sp}) \quad (5)$$

Where, S = saturation flow rate (pc/h/ln); S_O = base saturation flow rate (pc/h/ln); f_w = adjusted factor for lane width; f_{HVg} = adjusted factor heavy vehicle and grade; f_p = adjustment factor for the existence of a parking lane and activity; f_{bb} = adjustment factor for blocking effects of local buses; f_a = adjustment factor for area type; f_{Lu} = adjustment factor for lane utilization; f_{LT} = adjustment factor for left-turn vehicle presence in a lane group; f_{RT} = adjustment factor for right-turn vehicle presence in a lane group; f_{Lpb} = pedestrian adjustment factor for left-turn groups; f_{Rpb} = pedestrian–bicycle adjustment factor for right-turn groups; f_{wz} = adjustment factor for work zone presence at the intersection; f_{ms} = adjustment factor for downstream lane blockage; f_{sp} = adjustment factor for sustained spillback.

To split the dataset into training and testing subsets, a random sampling approach was employed, ensuring that both subsets captured the diverse traffic conditions and intersection geometries represented in the complete dataset. Specifically, 80% of the data were randomly selected for training, while the remaining 20% were used for testing and evaluating the models' performance on unseen data.

5. Delay models development

In this study, the ORANGE software package was used to preprocess, train, and evaluate the machine learning models. ORANGE is an open-source GUI (Graphic User Interface) platform that allows users to perform different supervised and supervised learning algorithms [75]. Fig. 7 shows a snapshot from ORANGE software.

Before developing any machine learning models, normalizing the feature data using a normalization technique is recommended. The normalization procedure is crucial for the accuracy and speed of the training phase. The purpose of this procedure is to standardize all data to the same scale, facilitating machine learning models in effectively optimizing model parameters and hyperparameters. The following equation represents the Z-score normalization procedure adopted in this study:

$$O = \frac{IN - Mean}{Std} \tag{6}$$

Where, O = the normalized output data point; IN = the un-normalized input data point; $Mean$ = the mean value of the selected feature in the dataset; Std = the standard deviation value of the selected feature in the dataset.

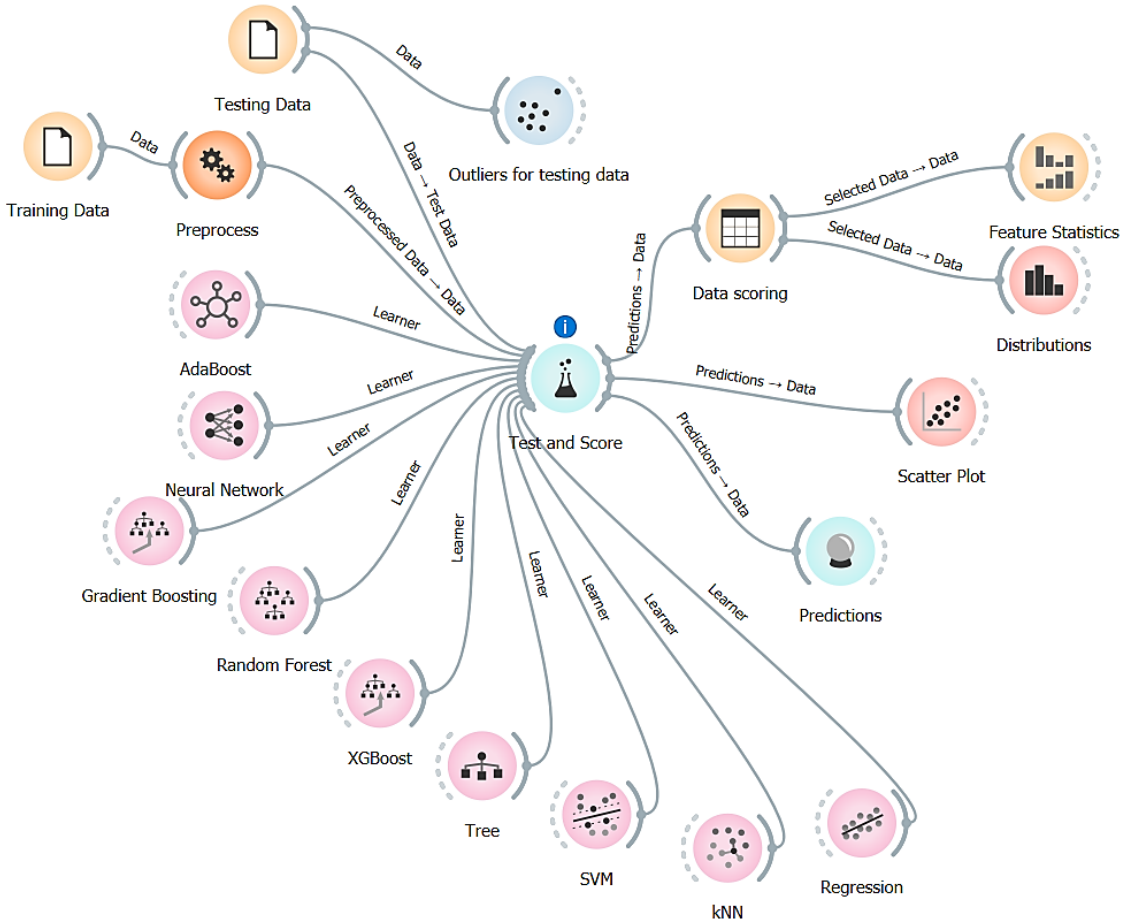

Fig. 7. A snapshot from ORANGE software.

In ORANGE software, the data were randomly divided into training and testing sets. The training set was utilized to build the machine learning models: AdaBoost, Gradient Boosting, XGBoost, Tree Decision, SVR, K-NN, and ANN. Subsequently, the testing set was employed to evaluate the performance of the machine learning models and measure their testing error. The data split ratios were 80% for training and 20% testing.

The study included a diverse set of machine learning algorithms, ranging from well-established techniques like Artificial Neural Networks, Random Forest, and Support Vector Machines to more recent methods such as XGBoost. This wide selection was designed to capture different modeling philosophies and their suitability for estimating vehicular delay at signalized intersections, building upon the limitations of prior studies that focused on a more limited set of algorithms. Fig. 8 illustrates the flowchart outlining the adopted methodology for preprocessing, training, and evaluating the machine learning models.

Fig. 8. A flowchart for the adopted methodology to test and evaluate the machine learning models.

5.1. The proposed tests

The study consists of five tests designed to evaluate the performance of the developed machine learning models. Each test and its respective objectives are outlined below:

5.1.1. Prediction accuracy

This test evaluates the performance of the machine learning models on both training and testing data by comparing predicted stopped delay values to actual values. Common measures such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Coefficient of Determination (R^2) are employed to assess error terms and goodness of fit. The following formulas represent the performance measures used to estimate the prediction accuracy of the machine learning models:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=0}^{n}(O-P)^2}{n}} \tag{7}$$

$$MAE\ (\%) = \frac{1}{n}\sum_{i=1}^{n}\left|\left(\frac{O-P}{O}\right)\right| \times 100 \tag{8}$$

$$R^2 = 1 - \frac{\sum_{i=0}^{n}(O-P)^2}{\sum_{i=0}^{n}(O-\bar{O})^2} \tag{9}$$

Where, $RMSE$ is the root mean square error; $MAE\ (\%)$ is the mean absolute relative error; R^2 is the coefficient of the determination; O = observed stopped delay (s/veh); P = the predicted stopped delay (s/veh); $\bar{O}$ is the average observed stopped delay (s/veh); n is the number of observed stopped delay points.

The $RMSE$ and MAE were used to determine the prediction ability of the models; the lower error indicates better prediction ability. R^2 was employed to o assess the goodness of fit for the developed models, with higher R^2 values indicating a better model fit to the data.

5.1.2. The performance gap between datasets

The purpose of this test is to assess the performance disparity between the developed machine learning models based on the RMSE error term calculated on both training and testing data. Given that the models were iteratively developed using various hyperparameters to minimize the gap between datasets, a significant difference in performance between training and testing sets may indicate overfitting to the training data. Moreover, if the training error substantially exceeds the testing error, it suggests a higher risk of poor performance on unseen data.

5.1.3. Models' sensitivity to outliers

Outliers are common in traffic engineering due to the inherent variability in the data and human factors. Therefore, modeling signalized vehicular delay must account for these outliers. In this paper, the authors aimed to estimate potential variations and inaccuracies in delay model inputs (arrival rate (V), effective green to cycle length (g/C), degree of saturation (X_s), and lane group width (W)). To assess the models' sensitivity to outliers, they first tested the models on the testing data without removing outliers. Subsequently, the models were retested on the testing

data after removing outliers. The difference in RMSE before and after outlier removal was calculated for each model, with a higher difference indicating greater sensitivity to outliers.

5.1.4. The model computational times for training and testing

The time required for training and testing machine learning models holds crucial importance for transportation agencies considering the adoption of such models. While updating machine learning models with new data is essential for achieving generalization, the process can be time-consuming, particularly with large training datasets. Thus, it is imperative to factor in the time needed for training and testing when comparing different models. In this study, computational times were measured using hardware specifications, including an Intel Core i7-8750H processor clocked at 2.2 GHz, 8 GB RAM, and a NVIDIA GTX 1050 graphics card. This information aids in understanding the feasibility and scalability of implementing these models in real-world applications.

5.1.5. The robustness of the developed models against data variation

Individual vehicle delays at signalized intersections exhibit wide variance due to the unpredictable nature of traffic arrivals and the interruption caused by traffic signal controls [76]. Despite the significant implications of such variance for signal control planning, design, and analysis [76], it is often overlooked in machine learning model development. Therefore, it is essential to test machine learning models on variations in traffic data to assess their robustness. In this study, synthetic data were generated using data augmentation techniques and tested alongside the testing dataset to ensure variability. Subsequently, the models were evaluated on these data to assess their robustness against data variation. These tests offer a comprehensive comparison of different machine learning algorithms from various perspectives, filling a gap in the literature where the focus has primarily been on measuring errors and goodness of fit without considering other aspects of model performance.

6. Results and discussions

6.1. Proposed models implementation

6.1.1. Regression model

Regression analysis was utilized in this study to formulate an empirical equation for predicting the average stopped delay at signalized intersections using traffic and geometry parameters. The dataset was divided into an 80% training set to calibrate the model and a 20% testing set to evaluate its performance. Partial Least Squares (PLS) regression with 10-fold cross-validation was employed instead of Ordinary Least Squares (OLS) regression to mitigate the risk of overfitting. Considering the variables listed in Table 1, the following PLS regression model was developed based on the logarithmic transformation of the dependent variable, which was found to be the most suitable for estimating the stopped delay:

$$\ln(D_s) = B_1\left(\frac{g}{C}\right) + B_2(X_s) + B_3(V) + B_4(W) + B_5 \quad (10)$$

$$D_s = e^{\left(B_1\left(\frac{g}{C}\right)+B_2(X_S)+B_3(V)+B_4(W)+B_5\right)} \quad (11)$$

Where, B_1 to B_5 = the regression coefficients; $\frac{g}{C}$ = effective green to cycle length ratio; X_s = degree of saturation; V = arrival rate (veh/h); W = approach width (m); D_s = the stopped delay per approach (s/veh).

The lowest error was observed at component number four. Subsequently, the regression coefficients were estimated, resulting in the final model as follows:

$$D_s = e^{\left(-4.647\left(\frac{g}{C}\right)+0.225(X_S)+0.001(V)-0.027(W)+4.380\right)} \quad (12)$$

With R^2 equal to 0.736 and RMSE equal to 16.771. The regression model was statistically significant at α- level < 5%. The model was tested on the testing data set, and the testing result is

Table 2
The training and testing result of the regression model.

Dataset	RMSE	MAE (%)	R^2
Training	16.771	23.725	0.736
Testing	24.137	26.402	0.903

shown in Table 2. The R^2 of the testing dataset was higher than the training set. However, the testing error was higher. The analysis of the developed regression model reveals that an increase in arrival rate and degree of saturation correlates with higher stopped delay. This is attributed to the resultant increase in queue length at the signalized intersection, as vehicles wait for the green phase. Conversely, a decrease in effective green ratio leads to higher stopped delay, as it extends the red time, resulting in more stopped vehicles. Furthermore, a decrease in approach width is associated with increased stopped delay at signalized intersections, stemming from the reduction in the number of lanes available.

The proposed model holds applicability for adoption in other developing countries sharing similar traffic behavior characteristics with Jordan. Alternatively, it can be calibrated to align with the specific traffic characteristics of these countries.

6.1.2. ANN model

Multiple ANN models were constructed, each with varying numbers of neurons in the hidden layer. Stochastic Gradient Descent (SGD) was employed as the learning algorithm, with the hyperbolic tangent (tanh) function serving as the activation function. The trial-and-error method was utilized to identify the optimal number of neurons in the hidden layer, guided by RMSE. Among these models, the 4-50-1 ANN configuration, featuring 50 neurons in the hidden layer, yielded the lowest error.

The performance of the developed model (4-50-1) in the training and testing data sets was investigated using Equations (4), (5), and (6). As seen in Table 3, the R^2 values are 0.943 and 0.947 in the training and testing sets, respectively. These results indicate a high correlation

between predicted values from the model and the observed values. Also, the RMSE values are 7.639 and 9.298 in the training and testing sets, respectively. The MAE values are 5.065 and 6.341 in the training and testing sets, respectively. This also indicates that the developed model performed well in all datasets.

Table 3
The training and testing results of the ANN model.

Dataset	RMSE	MAE (%)	R^2
Training	7.639	5.065	0.943
Testing	9.298	6.341	0.947

6.1.3. Tree decision and RF models

Tree regression and RF models were trained on the training dataset (80%) and evaluated on the testing set (20%). For the Tree model, the number of trees was limited to 100, and the minimum number of instances in leaves was set to 2. Through trial and error, the optimal number of tree nodes was determined to be 173, with 73 leaves.

In the case of the RF model, the number of trees (B) was selected using an error-based trial method, ensuring the lowest error rate for the corresponding number of trees. Additionally, the tree depth was restricted to 13 to prevent overfitting. Various RF models were tested with different numbers of predictors (N) and trees (B), with models featuring N=10 and B=5 demonstrating the highest prediction ability. The performance of the developed models was evaluated on both the training and testing datasets, as summarized in Table 4.

Table 4
The training and testing results of Tree and RF models.

Model	Dataset	RMSE	MAE (%)	R^2
Tree	Training	7.67	3.797	0.942
	Testing	9.421	5.078	0.946
RF	Training	6.698	3.734	0.956
	Testing	7.335	4.517	0.967

6.1.4. AdaBoost, gradient boost, and XGBoost models

The adopted algorithms were trained on the training dataset (80%) and evaluated on the testing set (20%). For AdaBoost, a learning rate of 0.5 was employed, and the optimal number of estimators (trees) producing the lowest error was determined to be 50 through trial and error. In the case of Gradient Boost, a learning rate of 0.1 was utilized, with the individual tree's maximum depth limited to 20 to prevent overfitting. The minimum splitting subset was set to 2, and the optimal number of trees was found to be 80 through trial and error.

For XGBoost, a learning rate of 0.1 was set, along with a maximum depth of 40 for individual trees. Regularization techniques were applied to mitigate overfitting, and the optimal number of trees was determined to be 50 through trial and error. The performance of the developed models was assessed on both the training and testing datasets, as summarized in Table 5.

6.1.5. SVM models

The data was split into 80% training set to train the model and 20% for testing. 10-fold cross-validation was used in order to determine the best model parameters. Different kernel functions (linear, RBF, sigmoid, and polynomial) were used in the model development phase in order to select the best kernel function that could be used with SVM to predict the average delay at signalized intersections.

10-fold cross-validation was used with trial and error approaches in order to select the best SVM model with its essential parameters such as kernel function, C parameter, and Є parameter. Table 6 shows the performance of the developed SVM models in the training and testing sets using different kernel functions. It was found that the RBF kernel function model had a higher prediction ability (R^2=0.921, MAE (%)=4.556, and RMSE=8.066 on the testing set) than other kernel functions with C =512 and Є = 0.1.

SVM with linear and Sigmoid kernels underperformed compared to Polynomial and RBF kernels due to their limitations in capturing nonlinear relationships and data distributions. The linear kernel assumes linear separability, which might not hold in complex datasets like stopped delay prediction at signalized intersections. Additionally, the Sigmoid kernel's sensitivity to scaling and its tendency to overfit can further exacerbate performance issues, leading to poorer generalization on unseen data.

Table 5
The training and testing result of boosting models.

Model	Dataset	RMSE	MAE (%)	R^2
AdaBoost	Training	4.216	2.122	0.983
	Testing	6.265	3.521	0.976
Gradient Boost	Training	4.555	2.888	0.980
	Testing	8.351	5.136	0.958
XGBoost	Training	5.192	2.596	0.974
	Testing	6.943	3.95	0.971

Table 6
The training and testing result of SVM models.

Model	Dataset	RMSE	MAE (%)	R^2
Linear	Training	17.996	9.132	0.682
	Testing	26.349	14.171	0.578
Polynomial	Training	9.073	5.247	0.919
	Testing	12.348	8.032	0.907
RBF	Training	8.066	4.556	0.936
	Testing	11.413	6.823	0.921
Sigmoid	Training	12.585	11.891	0.543
	Testing	30.431	17.591	0.438

6.1.6. K-NN model

The dataset was divided into two portions: 80% for training the model and the remaining 20% for both testing the developed K-NN model. By systematically trying different values of K, the optimal K value was determined through trial and error, selecting the model with the least error. The Mahalanobis distance was used in the K-NN training process. Notably, the K-NN model performed best when K was set to 10, resulting in RMSE of 12.149, MAE of 6.544% and R^2 of 0.855 on the training set. Furthermore, the model was tested on the testing set, achieving RMSE of 12.825, MAE of 7.623%, and R^2 of 0.900. These findings, summarized in Table 7, demonstrate the effectiveness of the K-NN model in both the training and testing datasets.

Table 7
The training and testing result of the K-NN model.

Dataset	RMSE	MAE (%)	R^2
Training	12.149	6.544	0.855
Testing	12.825	7.623	0.900

6.1.7. HCM, webster, and Akçelik models

The developed models, including regression, ANN, Tree, RF, SVM, K-NN, Gradient Boost, AdaBoost, and XGBoost, were compared with established analytical models such as HCM, Webster, and Akçelik models. This comparative analysis aimed to provide a practical assessment of their performance. The estimated delay values generated by these analytical models typically represent the total delay, also known as control delay. However, to focus specifically on stopped delay, the estimated delay values were adjusted using a correction factor of 1.3, as outlined in the HCM 2000-2010 guidelines [1,15]. This factor increases the stopped delay by 30% to account for deceleration and acceleration delays at signalized intersections. While useful for calibration, further field studies may be needed to refine its application. The testing results of the analytical models are summarized in Table 8. Additionally, Fig. 9 illustrates the estimated stopped delay plotted against the actual stopped delay values. The analysis reveals varying degrees of estimation accuracy among the models. Specifically, the Webster model demonstrates the poorest estimation performance, while the HCM model exhibits the most robust performance. This underscores the importance of selecting appropriate models for accurately estimating stopped delay at signalized intersections.

The HCM model achieved an RMSE of 32.639, with a MAE of 40.740% and an R^2 of 0.462 on the overall dataset. In contrast, the Webster model exhibited considerably higher error metrics, with an RMSE of 152.875, MAE of 102.994%, and R^2 of 0.176. The Akçelik model showed intermediate performance, with an RMSE of 31.072, MAE of 80.418%, and R^2 of 0.198. The low performance of analytical delay models (HCM, Webster, and Akcelik) underscore their limitations in accurately capturing the complex relationships between traffic variables and delay at signalized intersections. Despite their widespread use, these models fail to account for the nonlinear and intricate patterns present in real-world traffic data.

Table 8
The testing result of the adopted analytical models.

Model	Dataset	RMSE	MAE (%)	R^2
HCM	Overall	32.639	40.740	0.462
Webster	Overall	152.875	102.994	0.176
Akçelik	Overall	31.072	80.418	0.198

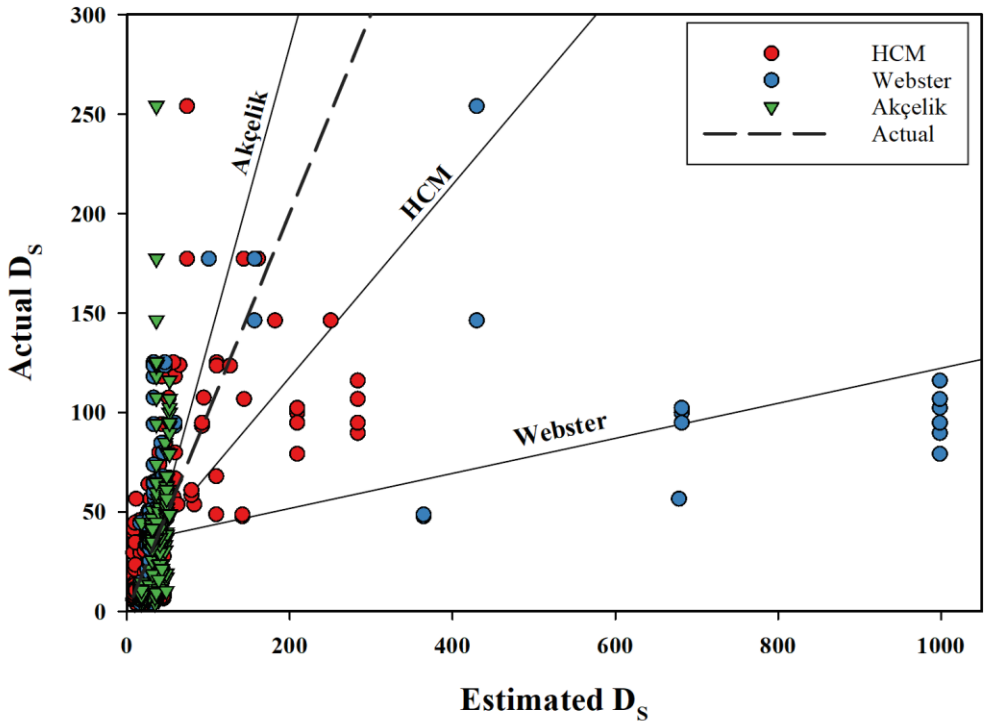

Fig. 9. Actual stopped delay vs predicted stopped delay using analytical models.

6.2. Comparisons between the developed models

The paper undertook comprehensive comparisons among the adopted models in terms of prediction accuracy, the performance gap between training and testing sets, sensitivity to outliers, computational time costs, and the robustness of the models. Fig. 10 illustrates the tests conducted and the respective procedures for each evaluation criterion. These comparisons offer insights into the effectiveness and reliability of the models in estimating stopped delay at signalized intersections.

6.2.1. Test 1 (prediction Accuracy)

The accuracy of predictions was assessed using both training and testing datasets, evaluating the developed models based on performance metrics outlined in Eq (7) – Eq (9). As shown in Fig. 11, regression and machine learning models exhibited significantly higher accuracy compared to analytical models. The RMSE values of analytical models were notably higher, ranging about 5 to 58 times higher than other models. Specifically, the Webster model emerged as the weakest performer, with an RMSE value of approximately 152.9.

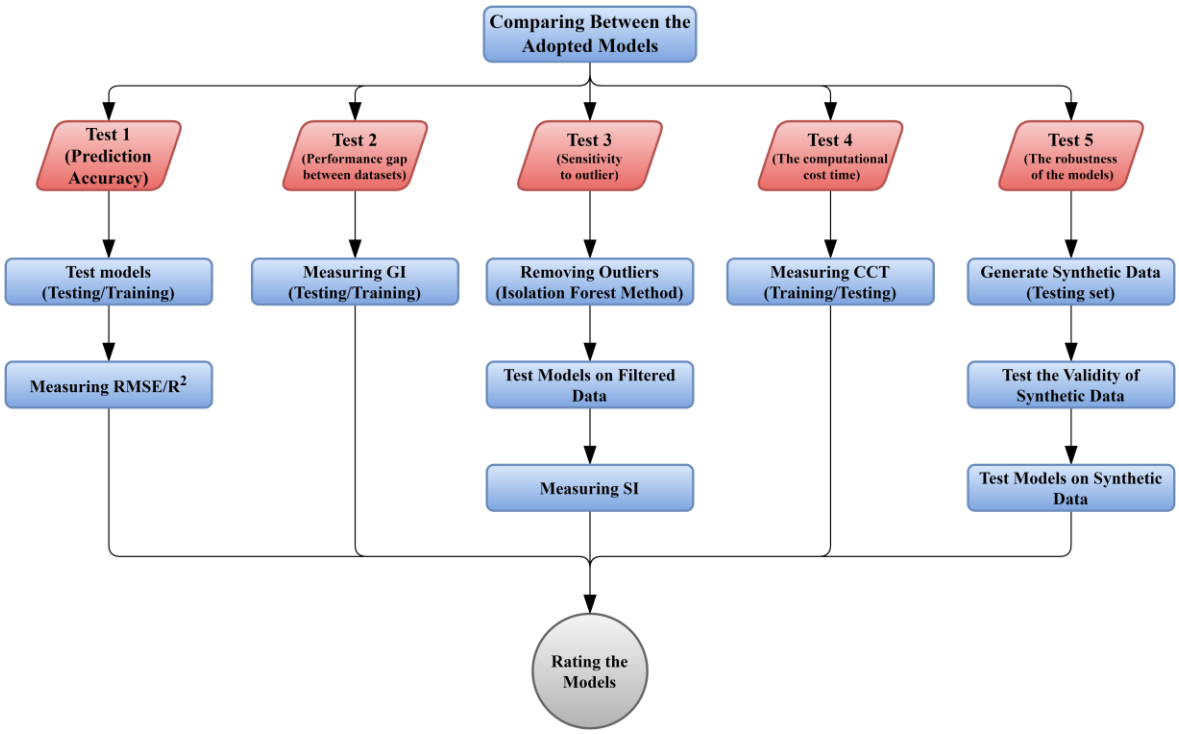
Comparing Between the Adopted Models
Test 1 (Prediction Accuracy)
Test 2 (Performance gap between datasets)
Test 3 (Sensitivity to outlier)
Test 4 (The computational cost time)
Test 5 (The robustness of the models)
Test models (Testing/Training)
Measuring GI (Testing/Training)
Removing Outliers (Isolation Forest Method)
Measuring CCT (Training/Testing)
Generate Synthetic Data (Testing set)
Measuring RMSE/R2
Test Models on Filtered Data
Test the Validity of Synthetic Data
Measuring SI
Test Models on Synthetic Data
Rating the Models

Fig. 10. The Methodology flowchart for the adopted tests in this study.

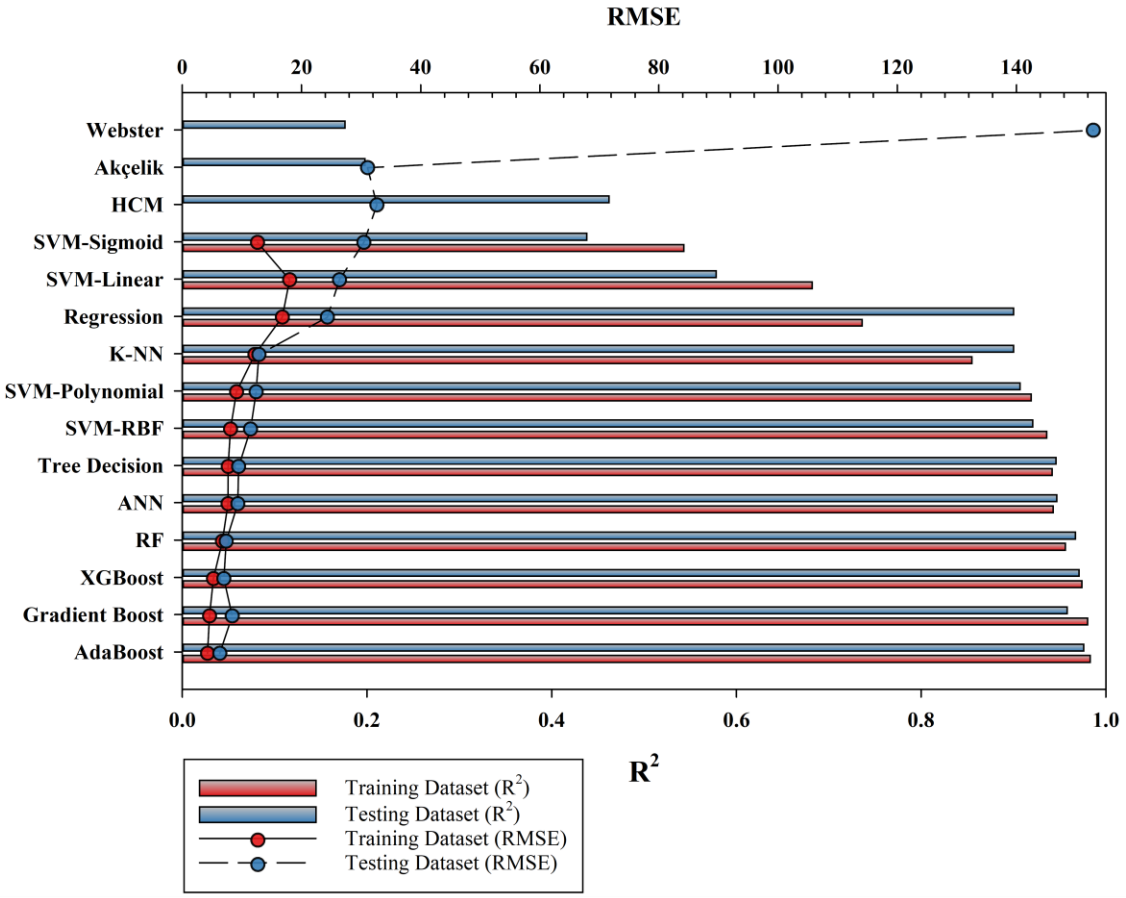

Fig. 11. The RMSE and R2 values for the analytical and machine learning models.

Conversely, machine learning models, except for SVM: Linear and Sigmoid, demonstrated superior prediction accuracy, with RMSE errors ranging from about 93% to 288% lower than the PLS regression model. Notably, Gradient Boost, XGBoost, AdaBoost, and RF models exhibited the most robust performance. Although other models displayed slightly higher RMSE margins (<13%), they still showcased commendable capabilities in predicting stopped delay.

Fig. 12 illustrates the RMSE and average predicted stopped delay values of the proposed models alongside actual values under varying saturation conditions. Analytical models (Webster, HCM, and Akçelik) tended to overestimate delay values when the degree of saturation exceeded 1, with Akçelik demonstrating the closest approximation to real values among the analytical models. Additionally, the developed regression model showed a propensity to overestimate delay values under similar conditions. In contrast, machine learning models demonstrated better prediction abilities across both under and over-saturation scenarios. Notably, as X_s increased, the accuracy of all models declined, particularly evident in regression and analytical models exhibiting poor performance at high X_s values. However, AdaBoost emerged as the most accurate model based on RMSE and Avg. D_s, having a relatively small RMSE of approximately 8 when X_s >1.

6.2.2. Test 2 (Performance gap between datasets)

In this study, the performance gap between training and testing sets was examined by comparing the models' performance on these two sets using the RMSE error term. It's important to note that the models underwent multiple iterations with different hyperparameters. Hyperparameter selection aimed to achieve the best performance and minimize the performance gap between training and testing sets. The performance gap index was calculated using the following formula:

$$GI = RMSE_{test} - RMSE_{train} \tag{13}$$

Where, GI= performance gap index; $RMSE_{train}$= $RMSE$ of the training dataset. $RMSE_{test}$ = $RMSE$ of the testing dataset. Fig. 13 shows the GI values for each of the developed models. It's clearly shown that the higher gap in training-testing performance was for the regression model. At the same time, the RF model revealed the lowest performance gap for training and testing sets.

6.2.3. Test 3 (Sensitivity to outlier)

The model sensitivity to outlier analysis was conducted on all models to identify the most robust model for outliers. This analysis is crucial to assess whether the models can handle challenging data. The same testing dataset was utilized for all models. Subsequently, the models were used to predict the stopped delay before and after removing the outliers. The prediction performance in terms of RMSE was then computed before and after removing the outliers. The following index was adopted to calculate the sensitivity to outliers for all models:

$$SI = RMSE_{before} - RMSE_{after} \tag{14}$$

Where, SI= sensitivity index; $RMSE_{before}$= $RMSE$ before removing the outliers from the testing data. $RMSE_{after}$ = $RMSE$ after removing the outliers from the testing data.

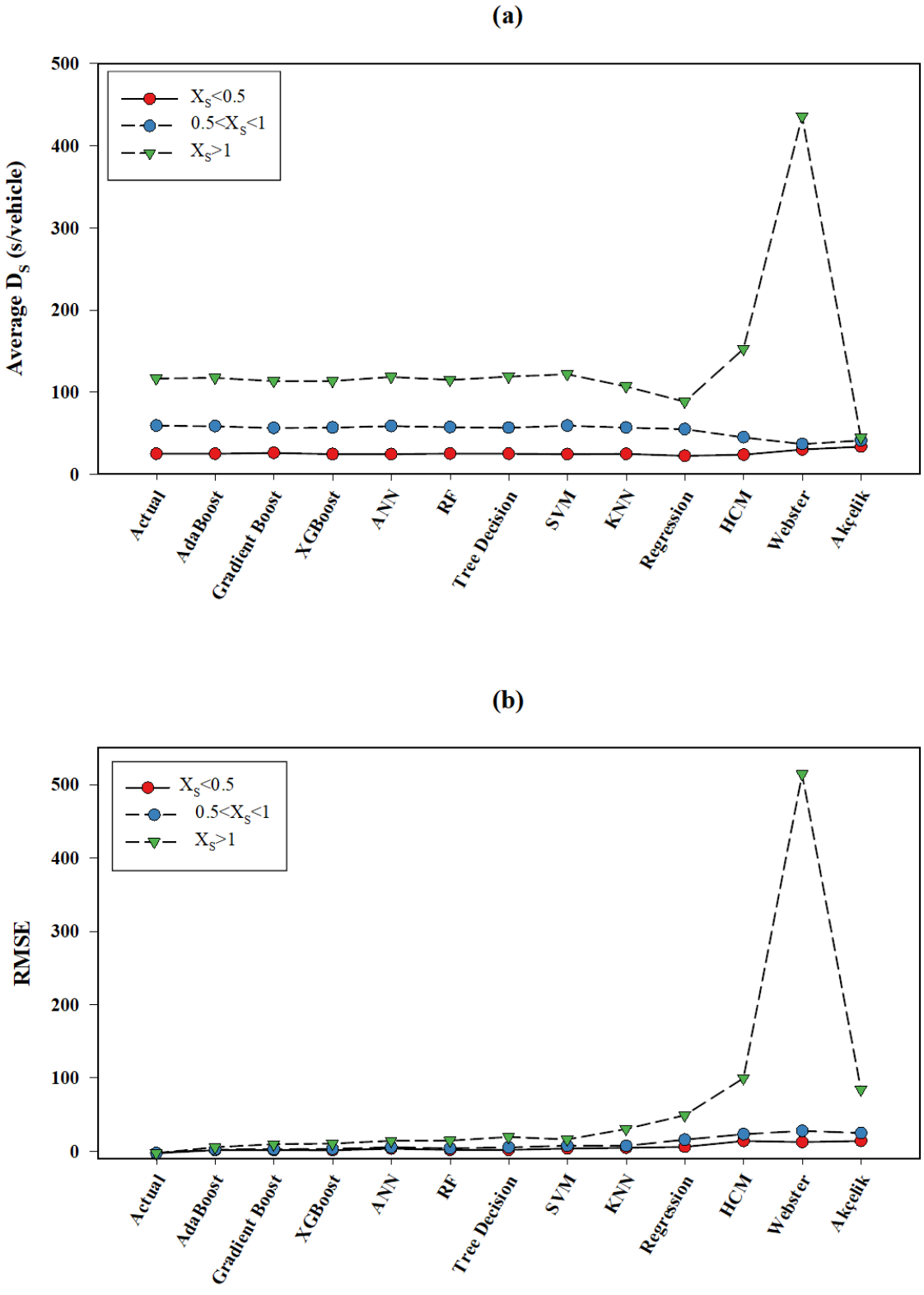

Fig. 12. (a) The average stopped delay values for each model under different saturation values. **(b)** The RMSE values for each model are under different saturation values.

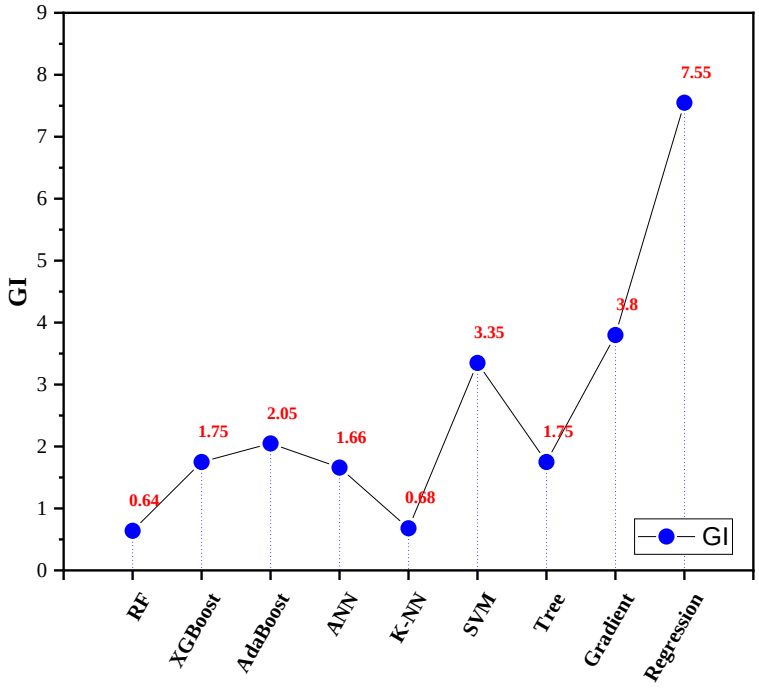

Fig. 13. The GI value for each machine learning model.

The higher the SI value, the greater the sensitivity to outliers. The Isolation Forest outlier detection method, implemented using ORANGE software, was employed to remove outliers (10%) from the test dataset. This method offers fast and effective outlier detection compared to other methods based on distance outliers' detection, which may have lower detection performance or require significantly more time [77].

Fig. 14 illustrates the RMSE values of each model after applying The Isolation Forest method and SI values. As shown in Fig. 14, higher RMSE and SI values were observed for the regression model, indicating a high sensitivity to outliers. Conversely, the RF model exhibited the lowest sensitivity to outliers based on RMSE and SI values.

6.2.4. Test 4 (The computational cost time)

The Computational Cost Time (CCT) of the adopted models was measured by the time required in seconds for the model to train on the training dataset and test on the testing dataset. The higher the time, the higher the cost of training and testing. Fig. 15 illustrates the CCT values for each developed model. The most time-intensive model was ANN, with an average testing and training time of 6.66 seconds. Conversely, the least time-intensive model in terms of CCT was regression, owing to the lower complexity of the regression approach compared to machine learning algorithms.

6.2.5. Test 5 (The robustness of the model)

The data augmentation technique was employed to assess the robustness of the machine learning models. Robustness, in this context, refers to the stability of the models' prediction accuracy amidst variations in the data. One common challenge with machine learning models is overfitting to the training data.

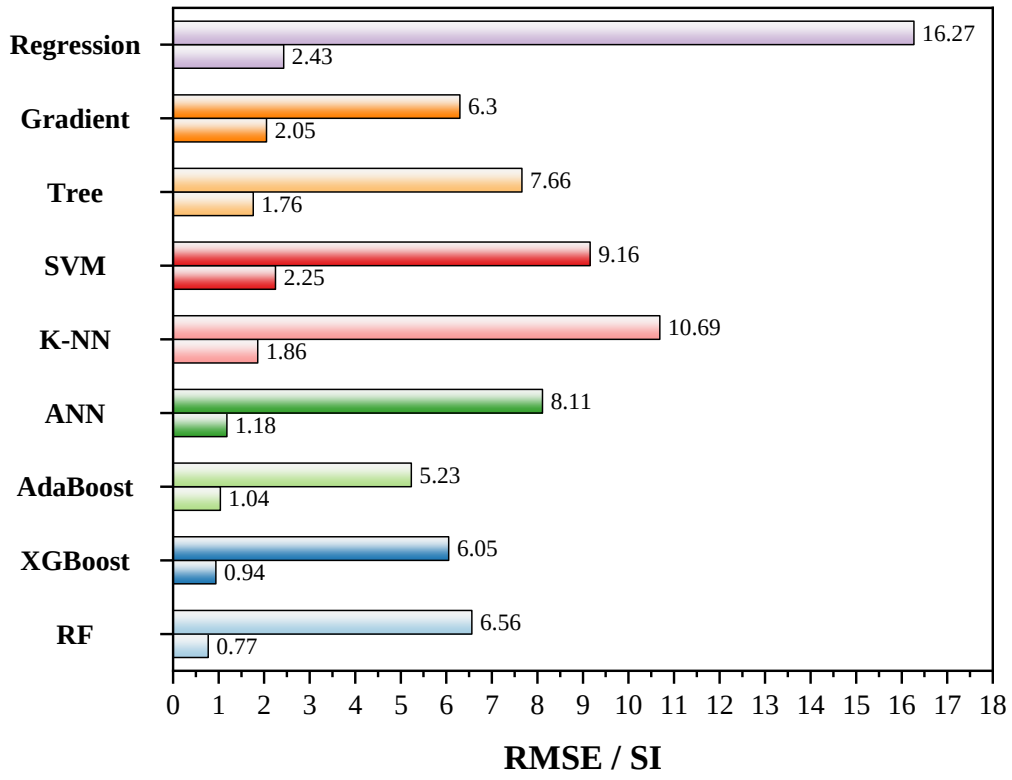

Fig. 14. RMSE and SI values for each machine learning model.

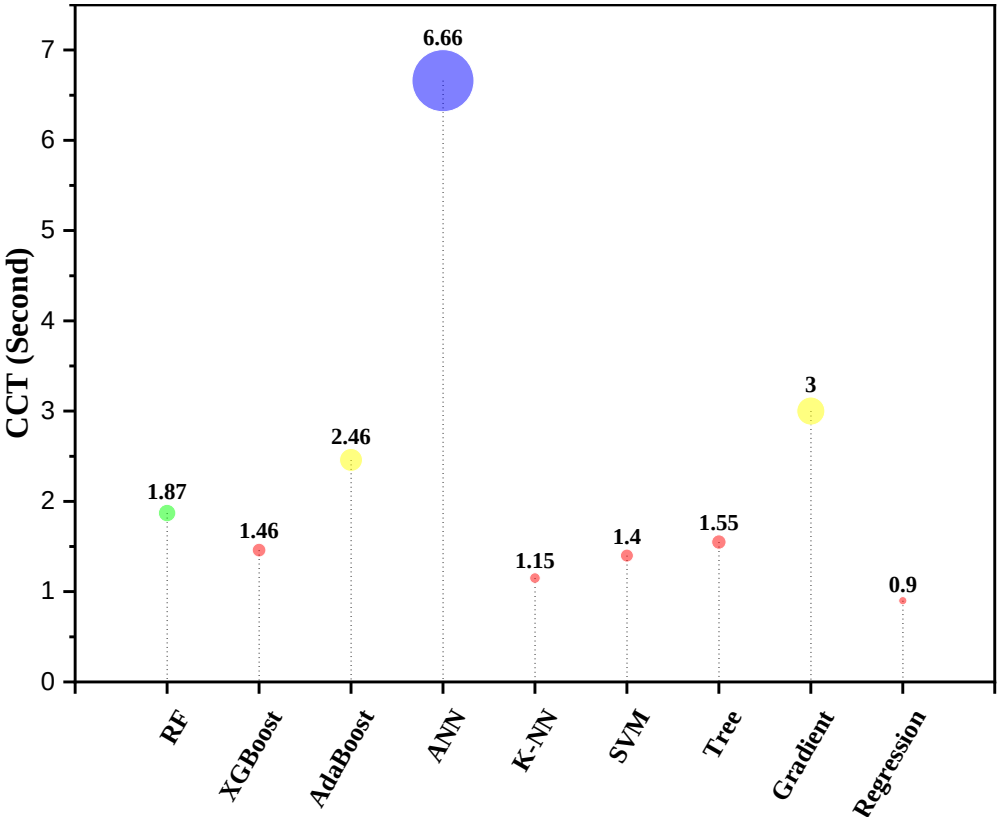

Fig. 15. The CCT value for each machine learning model.

To evaluate the robustness of the proposed models, synthetic data were generated based on the original testing dataset. The Pysynth [78] library package for Python Programming Language was utilized to create synthetic data using the Iterative Proportional Fitting (IPF) approach. This approach constructs synthetic data with the same statistical characteristics and distributions as the original data, though with entirely distinct data points, enabling a comprehensive examination of model robustness.

This package was utilized to generate ten distinct synthetic datasets from the testing dataset. The reliability of these synthetic datasets was assessed based on their statistical characteristics in comparison to the original testing data. Fig. 16 illustrates the statistical features and distribution of D_s values for both the original and synthetic data. Additionally, Pearson correlations between variables were examined to evaluate the similarity between the synthetic and original testing data.

Fig. 17 shows an example of the correlation between variables in the synthetic and original datasets, revealing minimal absolute correlation differences. Subsequently, each dataset was individually fed into the machine learning models to predict stopped delays. The prediction accuracy, measured in terms of RMSE for each model in every dataset, was averaged across the ten datasets to calculate the final RMSE using the following formula:

$$RMSE_n = \frac{RMSE_{n1} + \cdots RMSE_{n10}}{10} \quad (15)$$

Where, $RMSE_n$= Root Mean Square of machine learning model n.

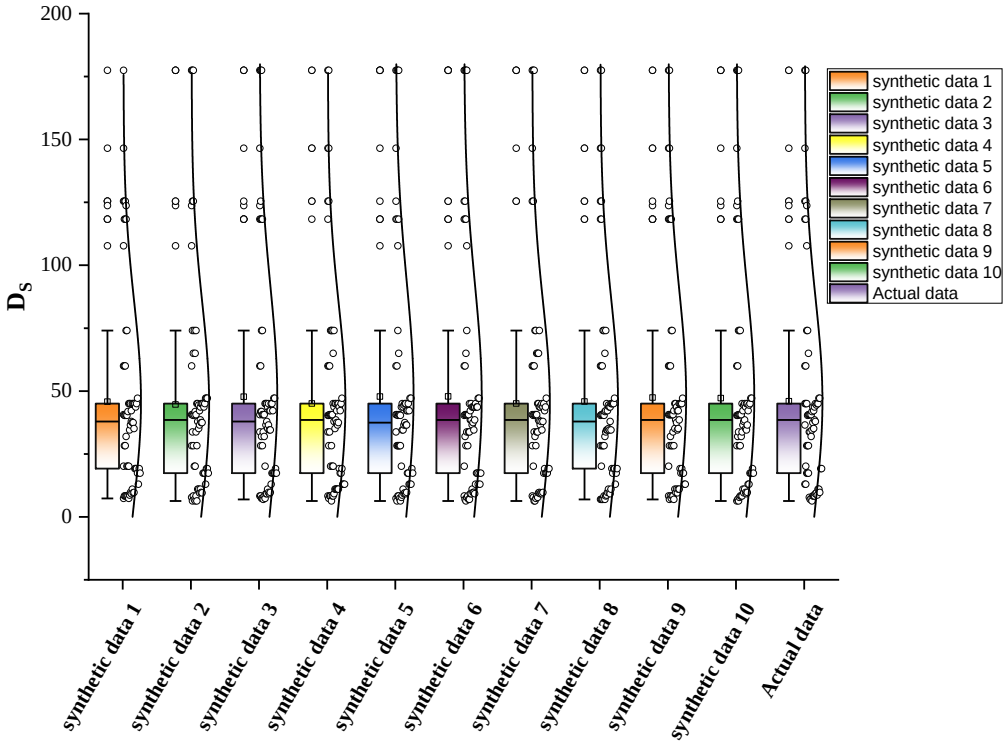

Fig. 16. The statistical characteristics and distribution of Ds values for the original and synthetic data.

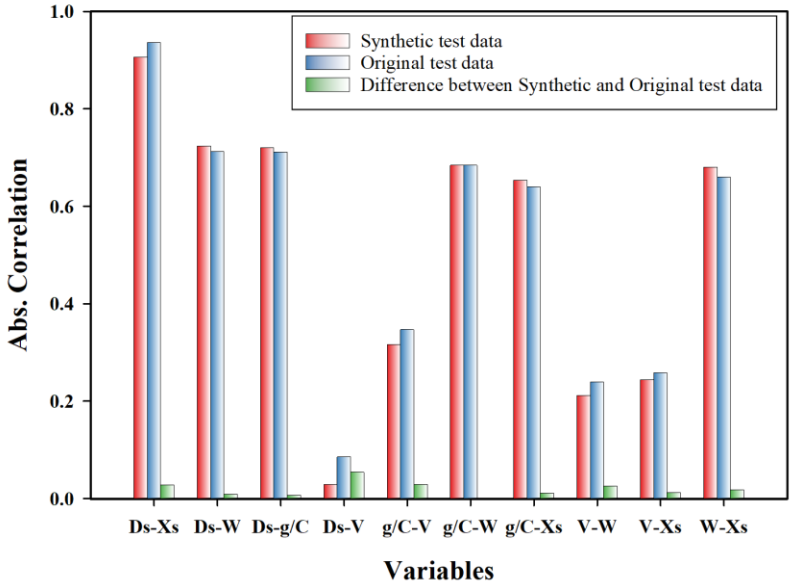

Fig. 17. the correlation between variables of a synthetic and the original data.

Fig. 18 shows the min, max, and average RMSE values in each dataset corresponding to each model. The result demonstrated that the ANN model had the highest robustness and stability against the variation in the data, with average RMSE values equal to 15.47. The highest average RMSE value was found for the regression model, revealing the lowest stability against variation of the data.

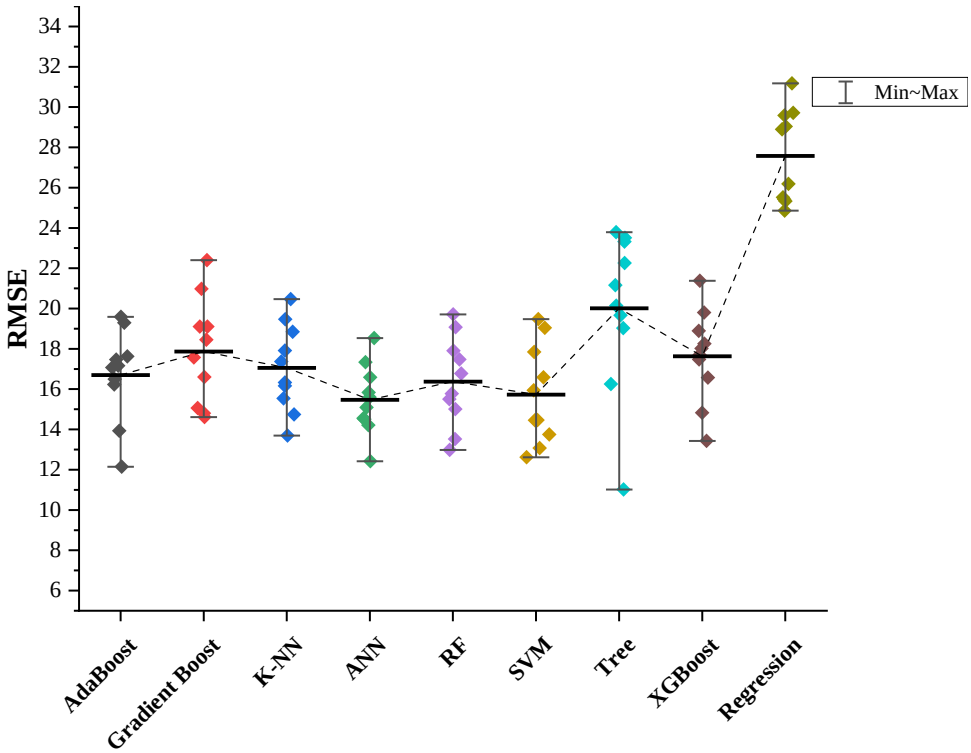

Fig. 18. the min, max, and average RMSE values in each synthetic dataset for each model.

6.3. Rating the models

Table 9 presents a comparative analysis of all models utilized in this study, considering key criteria such as prediction accuracy, performance gap between training and testing sets, sensitivity to outliers, and computational time costs. Each model received a rating from 1 to 9 for each criterion, with a higher rating indicating superior performance. Total ratings for each model were computed by summing its ratings across all criteria, and models were then ranked accordingly. In cases of equal ratings, models were prioritized based on overall accuracy. AdaBoost appeared as the top-ranked model for prediction accuracy, showing an average error of under 2.5% (RMSE=4.7). Conversely, the regression model exhibited the lowest prediction accuracy, with an average error of approximately 25% (RMSE=18.7).

Regarding the performance gap between training and testing, the RF model ranked highest, displaying a minimal GI value of about 0.64, while the regression model ranked lowest, indicating the widest performance gap with a GI value of approximately 2.43.

Table 9
The results of comparison between all machine learning models.

Rank	Model	Test 1		Test 2		Test 3		Test 4		Test 5		Total Rating
		Val.	Rat.	Val.	Rat.	Val.	Rat.	Val.	Rat.	Val.	Rat.	
1	RF	6.83	6	0.64	9	0.77	9	1.87	4	16.37	7	35
2	XGBoost	5.59	7	1.75	6	0.94	8	1.46	6	17.64	4	31
3	AdaBoost	4.70	9	2.05	4	1.04	7	2.46	3	16.7	6	29
4	ANN	8.00	5	1.66	7	1.18	6	6.66	1	15.47	9	28
5	K-NN	12.29	2	0.68	8	1.86	4	1.15	8	17.06	5	27
6	SVM	8.84	3	3.35	3	2.25	2	1.40	7	15.73	8	23
7	Tree	8.05	4	1.75	5	1.76	5	1.55	5	20.02	2	21
8	Gradient Boost	5.45	8	3.80	2	2.05	3	3.00	2	17.87	3	18
9	Regression	18.70	1	7.55	1	2.43	1	0.90	9	27.57	1	13

Val. = Value, Rat. = Rating

For sensitivity to outliers, the RF model was the lowest, with an SI value of about 0.77, and was ranked first among other models. Again, the regression model was ranked the latest in terms of sensitivity for outliers, with a high SI value of about 2.43 compared to other models.

In terms of CCT, the regression model revealed the lowest training and testing time compared to other models, although it had the lowest prediction accuracy. However, the ANN model was the most time-consuming algorithm among other models, although it had a high prediction performance.

In terms of model robustness to variation of data, the ANN model revealed the highest stability of the predictions and lowest sensitivity to variation in data. On the other hand, the regression model demonstrated the lowest stability, which is expected due to the nature of the regression analysis approach.

The results also indicated that while Gradient Boost demonstrated very high prediction accuracy, ranking second overall in both training and testing sets compared to other models, it tended to

overfit the training data. Moreover, it showed high sensitivity to outliers and required significant time for training and testing. Additionally, the model showed instability in scenarios involving data variation.

Finally, according to the comparison conducted, as shown in Table 9, the most balanced machine learning model and the highest based on the criteria was the RF model. The total rating of RF was higher than other models (35 out of 45). On the other hand, the lowest rating was the regression model (13 out of 45).

7. Conclusions and recommendations

7.1. Conclusions

This research presents a comprehensive comparison between various machine learning algorithms for estimating stopped delay models at signalized intersections and for application in design and planning contexts. Eight signalized intersections in Amman, the capital of Jordan, were utilized as a case study. The following conclusions can be drawn from this work:

1- Eight machine learning algorithms were implemented, including AdaBoost, Gradient Boosting, XGBoost, decision tree, support vector machine (SVM), k-nearest neighbors (K-NN), and artificial neural network (ANN). All models succeeded in estimating stopped delay at signalized intersections using only four input variables - arrival rate (V), effective green time to cycle length ratio (g/C), degree of saturation (X), and approach width (W) - achieving average errors between approximately 2% and 7%.

2- To benchmark machine learning techniques, the Highway Capacity Manual (HCM), Webster, and Akçelik models, prevalent conventional methods in the literature, were adopted. The Analytical delay models were found to have very high RMSE values, approximately 5-58 times higher than machine learning models, highlighting their inaccuracy when used without calibration for the study context. Machine learning models demonstrated higher accuracy than regression and analytical models for both under-saturated and over-saturated conditions.

3- The comparison was based on five criteria - prediction accuracy, performance divergence between training and testing, sensitivity to outliers, computational time cost, and robustness against data variation - since most prior studies, to our knowledge, only examined accuracy. The Random Forest algorithm emerged as the most balanced and robust model, achieving an average error of less than 4% while also performing well across other evaluation criteria like training-testing gap, outlier sensitivity, and computational efficiency.

4- For prediction accuracy (Test 1) on the test dataset, the AdaBoost model achieved the highest ranking, closely followed by Gradient boosting and XGBoost. For the lowest gap between the training and testing set, the RF model performed better than other models, followed by K-NN and ANN. For robustness to outliers (Test 3), the RF was the top-ranked model, followed by AdaBoost and XGBoost. The regression method exhibited the fastest

computational training time (Test 4), ranking highest for efficiency, followed by K-NN and SVM. In terms of robustness to variation (Test 5), the ANN model performed the best by ranking, followed by SVM and RF.

5- The PLS regression model has the lowest total rating of 13 out of 45. While RF showed the best total ratting of 35 out of 45, followed by XGBoost and AdaBoost.

6- Despite higher accuracy, machine learning methods require longer training and calculation times than conventional techniques. Thus, time cost is an important consideration for transportation agencies.

This study makes significant contributions to advancing the existing knowledge in traffic engineering and signalized intersection delay modeling. Firstly, it establishes machine learning techniques, as substantially superior to traditional analytical delay models like HCM, Webster, and Akcelik, which exhibited large inaccuracies when applied without recalibration. The comprehensive comparative evaluation framework across prediction accuracy, training-testing gap, outlier sensitivity, computational cost, and robustness provides traffic engineers a robust methodology to select the most appropriate algorithm based on project priorities. Top performing algorithms like Random Forest, XGBoost, and AdaBoost are identified, providing insights into their relative strengths for delay estimation. Notably, by utilizing a diverse dataset spanning under-saturated and over-saturated conditions, the machine learning models developed enhance generalizability and real-world relevance. Overall, this work advocates a shift towards data-driven approaches that can learn accurate delay predictions from data, rather than being constrained by simplifying assumptions of conventional models. It pushes the boundaries of signalized intersection modeling towards more accurate data-driven techniques.

7.2. Limitations and recommendations

While this study provides insights into the performance of different machine learning algorithms for delay estimation, there are several potential areas for future research:

1. Prediction accuracy was evaluated using average error metrics, which may not fully capture the variability or distribution of errors across different traffic conditions or intersection characteristics. Thus, future work could explore alternative accuracy metrics beyond average errors, such as quantile loss functions or distribution-based measures, to better capture the variability and distribution of prediction errors.

2. The divergence between training and testing performance was assessed, but the specific train-test split and data sampling techniques used could influence these results. Consequently, investigating the impact of different train-test split techniques and data sampling strategies on the divergence between training and testing performance is recommended, aiming to develop robust and generalizable models.

3. Sensitivity to outliers was considered, but the definition and identification of outliers may vary depending on the specific application context and data characteristics. To address this, establishing standardized methods for identifying and handling outliers in the context of

delay estimation would be beneficial, as the sensitivity to outliers may vary depending on the specific application and data characteristics.

4. The computational time cost was measured, but the hardware specifications and implementation details could impact these timings, making direct comparisons across studies challenging. Therefore, developing benchmarking frameworks and standardized datasets could enable more consistent and comparable evaluations of computational time costs across different hardware platforms and implementation details.
5. Model robustness was evaluated based on variations in the input data, but other factors, such as changes in traffic patterns, signal timing plans, or external conditions, were not considered. Accordingly, extending the robustness analysis to consider a broader range of factors that may influence model performance, such as changes in traffic patterns, signal timing plans, or external conditions (e.g., weather, incidents, special events), is advisable.
6. The comparison focused on a specific set of criteria, but other important factors, such as interpretability, scalability, or ease of deployment, were not explicitly evaluated. Hence, incorporating additional evaluation criteria that are relevant to transportation agencies, such as model interpretability, scalability, ease of deployment, and integration with existing traffic management systems, would be valuable.

Funding

This research received no external funding.

Conflicts of interest

The authors declare no conflict of interest.

Authors' contributions statement

Y.I. Alatoom: Data analysis, interpreting results, writing - original draft, investigation, methodology, resources, validation, visualization. A.B. Al-Hamdan: Writing - original draft, writing - review & editing.

References

[1] HCM. Highway Capacity Manual. Washington, D.C.: Transportation Research Board; 2010.

[2] Kang YS. Delay, stop and queue estimation for uniform and random traffic arrivals at fixed-time signalized intersections. Virginia Polytechnic Institute and State University, 2000.

[3] Quiroga CA, Bullock D. Measuring Control Delay at Signalized Intersections. J Transp Eng 1999;125:271–80. https://doi.org/10.1061/(ASCE)0733-947X(1999)125:4(271).

[4] Dion F, Rakha H, Kang Y-S. Comparison of delay estimates at under-saturated and over-saturated pre-timed signalized intersections. Transp Res Part B Methodol 2004;38:99–122. https://doi.org/10.1016/S0191-2615(03)00003-1.

[5] Webster VF. Traffic Signal Settings. Road Research Technical Paper No. 39. Berkshire, England: Road Research Laboratory, Her Majesty's Stationery Office; 1958.

[6] HCM. Highway Capacity Manual. Washington, D.C.: Transportation Research Board; 1985.

[7] HCM. Highway Capacity Manual. Washington, D.C.: Transportation Research Board; 1992.

[8] Akcelik R. The Highway Capacity Manual Delay Formula for Signalized Intersections. ITE J 1988;58:23–7.

[9] HCM. Highway Capacity Manual. Washington, D.C.: Transportation Research Board; 1981.

[10] Akcelik R. TRAFFIC SIGNALS: CAPACITY AND TIMING ANALYSIS. Report No. 123. Australian Road Research Board; 1981.

[11] Richardson D, Schnablegger J, Stephenson B, Teply S. Canadian capacity guide for signalised intersections. Edmonton, Canada: Institute of Transportation Engineers, District 7, University of Alberta; 1984.

[12] Thamizh Arasan V, Jagadeesh K. Effect of Heterogeneity of Traffic on Delay at Signalized Intersections. J Transp Eng 1995;121:397–404. https://doi.org/10.1061/(ASCE)0733-947X(1995)121:5(397).

[13] Ahmed K, Abu-Lebdeh G, Al-Omari B. Estimation of Delay Induced by Downstream Operations at Signalized Intersections over Extended Control Time. J Transp Eng 2013;139:8–19. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000465.

[14] Korkmaz E, Akgüngör AP. Delay estimation models for signalized intersections using differential evolution algorithm. J Eng Res 2017;5:16–29.

[15] HCM. Highway Capacity Manual. Washington, D.C.: Transportation Research Board; 2000.

[16] Sofia GG, Kadhim AJ, Khalil SM. Development of Delay Model for Selected Signalized Intersections at CBD in Sulaymaniyah City. J Eng Dev 2014;18.

[17] Safaeian Hamzehkolaei N, Alizamir M. Performance evaluation of machine learning algorithms for seismic retrofit cost estimation using structural parameters. J Soft Comput Civ Eng 2021;5:32–57. https://doi.org/10.22115/SCCE.2021.284630.1312.

[18] Vatani A, Jafari-Asl J, Ohadi S, Safaeian Hamzehkolaei N, Afzali Ahmadabadi S, Correia JAFO. An efficient surrogate model for reliability analysis of the marine structure piles. Proc Inst Civ Eng - Marit Eng 2023;176:176–92. https://doi.org/10.1680/jmaen.2022.020.

[19] Chen L, Fakharian P, Rezazadeh Eidgahee D, Haji M, Mohammad Alizadeh Arab A, Nouri Y. Axial compressive strength predictive models for recycled aggregate concrete filled circular steel tube columns using ANN, GEP, and MLR. J Build Eng 2023;77:107439. https://doi.org/10.1016/j.jobe.2023.107439.

[20] Zhang G, Safaeian Hamzehkolaei N, Rashnoozadeh H, Band SS, Mosavi A. Reliability assessment of compressive and splitting tensile strength prediction of roller compacted concrete pavement: introducing MARS-GOA-MCS. Int J Pavement Eng 2022;23:5030–47. https://doi.org/10.1080/10298436.2021.1990920.

[21] Siahkali MZ, Ghaderi A, Bahrpeyma A, Rashki M, Safaeian Hamzehkolaei N. Estimating Pier Scour Depth : Comparison of Empirical Formulations with ANNs, GMDH, MARS, and Kriging. J AI Data Min 2021;9:109–28. https://doi.org/10.22044/jadm.2020.10085.2147.

[22] Ashrafian A, Safaeian Hamzehkolaei N, Dwijendra NKA, Yazdani M. An Evolutionary Neuro-Fuzzy-Based Approach to Estimate the Compressive Strength of Eco-Friendly Concrete Containing Recycled Construction Wastes. Buildings 2022;12:1280. https://doi.org/10.3390/buildings12081280.

[23] Tabatabai SM, Tabatabai FA-S. Integrating Inverse Data Envelopment Analysis and Machine Learning for Enhanced Road Transport Safety in Iran. J Soft Comput Civ Eng 2024;8:141–60. https://doi.org/10.22115/scce.2023.399661.1657.

[24] Sharmila RB, Velaga NR. Mode-Wise Corridor Level Travel-Time Estimation Using Machine Learning Models. J Soft Comput Civ Eng 2020;4:72–97. https://doi.org/10.22115/scce.2020.215679.1164.

[25] Asencio-Cortés G, Florido E, Troncoso A, Martínez-Álvarez F. A novel methodology to predict urban traffic congestion with ensemble learning. Soft Comput 2016;20:4205–16. https://doi.org/10.1007/s00500-016-2288-6.

[26] Qiao F, Yi P, Yang H, Devarakonda S. Fuzzy logic based intersection delay estimation. Math Comput Model 2002;36:1425–34. https://doi.org/10.1016/S0895-7177(02)00298-4.

[27] Sazi Murat Y, Baskan Ö. Modeling vehicle delays at signalized junctions: Artificial neural networks approach. J Sci Ind Res (India) 2006;65:558–64.

[28] Preethi P, Varghese A, Ashalatha R. Modelling Delay at Signalized Intersections under Heterogeneous Traffic Conditions. Transp Res Procedia 2016;17:529–38. https://doi.org/10.1016/j.trpro.2016.11.107.

[29] Dogan E, Payidar Akgungor A, Arslan T. Estimation of delay and vehicle stops at signalized intersections using artificial neural network. Eng Rev 2016;36:157–65.

[30] Klir GJ, Yuan B. Fuzzy Sets and Fuzzy Logic: Theory and Applications. Pearson; 2015.

[31] Zadeh LA. Fuzzy sets. Inf Control 1965;8:338–53.

[32] Zadeh L. Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets Syst 1978;1:3–28.

[33] Al-Fahdawi OAS, Barroso LR, Soares RW. Adaptive Neuro-Fuzzy and Simple Adaptive Control Methods for Attenuating the Seismic Responses of Coupled Buildings with Semi-Active Devices: Comparative Study. J Soft Comput Civ Eng 2019;3:1–21. https://doi.org/10.22115/scce.2019.199731.1128.

[34] Hasiloglu A, Gokdag M, Karsli N. Comparison an artificial intelligence-based model and other models: signalized intersection delay estimates. J Eng Innov Technol 2014;4:220–31.

[35] Gökdağ M, Haşıloğlu A, Karslı N, Atalay A, Akbas A. Modeling of vehicle delays at signalized intersection with an adaptive neuro-fuzzy (ANFIS). J Sci Ind Res (India) 2007;66.

[36] Sazi Murat Y. Comparison of fuzzy logic and artificial neural networks approaches in vehicle delay modeling. Transp Res Part C Emerg Technol 2006;14:316–34. https://doi.org/10.1016/j.trc.2006.08.003.

[37] Najafi Moghaddam Gilani V, Ghanbari Tamrin MR, Hosseinian SM, Nikookar M, Safari D, YektaParast S. Investigation of Bus Special Lane Performance Using Statistical Analysis and Optimization of the Signalized Intersection Delay by Machine Learning Methods. J Optim 2022;2022:1–24. https://doi.org/10.1155/2022/2984803.

[38] Mahmoud N, Abdel-Aty M, Cai Q, Yuan J. Predicting cycle-level traffic movements at signalized intersections using machine learning models. Transp Res Part C Emerg Technol 2021;124:102930. https://doi.org/10.1016/j.trc.2020.102930.

[39] Ranpura P, Shukla V, Gujar R. Estimation of vehicle control delay using artificial intelligence techniques for heterogeneous traffic conditions. Expert Syst Appl 2024;246:123206. https://doi.org/10.1016/j.eswa.2024.123206.

[40] Bagdatli MEC. Vehicle Delay Modeling at Signalized Intersections with Gene-Expression Programming. J Transp Eng Part A Syst 2020;146:04020107. https://doi.org/10.1061/JTEPBS.0000423.

[41] Bagdatli MEC, Dokuz AS. Vehicle Delay Estimation at Signalized Intersections Using Machine Learning Algorithms. Transp Res Rec J Transp Res Board 2021;2675:110–26. https://doi.org/10.1177/03611981211036874.

[42] Priddy KL, Keller PE. Artificial Neural Networks: An Introduction. SPIE Press; 2005.

[43] Alatoom YI, Al-Suleiman (Obaidat) TI. Development of pavement roughness models using Artificial Neural Network (ANN). Int J Pavement Eng 2022;23:4622–37. https://doi.org/10.1080/10298436.2021.1968396.

[44] Arifuzzaman M. Advanced ANN Prediction of Moisture Damage in CNT Modified Asphalt Binder. J Soft Comput Civ Eng 2017;1:1–11. https://doi.org/10.22115/scce.2017.46317.

[45] Faghri A, Hua J. EVALUATION OF ARTIFICIAL NEURAL NETWORK APPLICATIONS IN TRANSPORTATION ENGINEERING. Report No. 1358. Washington, D.C.: Transportation Research Board; 1992.

[46] Murat YS, Ceylan H. Use of artificial neural networks for transport energy demand modeling. Energy Policy 2006;34:3165–72. https://doi.org/DOI: ,.

[47] Wang J, Chen J, Ouyang S, Yang Y. Trajectory tracking control based on adaptive neural dynamics for four-wheel drive omnidirectional mobile robots. Eng Rev 2014;34:235–43.

[48] Karayiannis N, Venetsanopoulos A. Artificial neural networks. Boston: Kluwer Academic; 1993.

[49] Shanmuganathan S, Samarasinghe S. Artificial Neural Network Modelling. Springer International Publishing; 2016.

[50] Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and Regression Trees. Taylor \& Francis; 1984.

[51] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. Springer New York; 2009.

[52] Breiman L. Random Forest. Mach Learn 2001;45:5–32.

[53] George T, Ganesan V. Advanced cuttlefish optimizer-random decision forest (ACORDF) based design of fractional order PID controller for higher-order time-delay system. J Eng Des Technol 2021;ahead-of-p. https://doi.org/10.1108/JEDT-11-2020-0451.

[54] Dutta RK, Gnananandarao T, Sharma A. Application of Random Forest Regression in the Prediction of Ultimate Bearing Capacity of Strip Footing Resting on Dense Sand Overlying Loose Sand Deposit. J Soft Comput Civ Eng 2019;3:28–40. https://doi.org/10.22115/scce.2019.137910.1080.

[55] Ho TK. Random Decision Forests. Proc. Third Int. Conf. Doc. Anal. Recognit. (Volume 1) - Vol. 1, USA: IEEE Computer Society; 1995, p. 278.

[56] Ho TK. The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 1998;20:832–44. https://doi.org/10.1109/34.709601.

[57] Mitchell TM. Machine Learning. McGraw-Hill Education; 1997.

[58] Gong H, Sun Y, Shu X, Huang B. Use of random forests regression for predicting IRI of asphalt pavements. Constr Build Mater 2018;189:890–7. https://doi.org/10.1016/j.conbuildmat.2018.09.017.

[59] Chen P, Sun J, Qi H. Estimation of delay variability at signalized intersections for urban arterial performance evaluation. J Intell Transp Syst 2017;21:94–110. https://doi.org/10.1080/15472450.2016.1216319.

[60] Kégl B. The return of AdaBoost.MH: multi-class Hamming trees 2013.

[61] Long T, He B, Ghorbani A, Khatami SMH. Tree-Based Techniques for Predicting the Compression Index of Clayey Soils. J Soft Comput Civ Eng 2023;7:52–67. https://doi.org/10.22115/scce.2023.377601.1579.

[62] Boser BE, Guyon IM, Vapnik VN. A Training Algorithm for Optimal Margin Classifiers. Proc. Fifth Annu. Work. Comput. Learn. Theory, New York, NY, USA: Association for Computing Machinery; 1992, p. 144–152. https://doi.org/10.1145/130385.130401.

[63] Brereton RG, Lloyd GR. Support Vector Machines for classification and regression. Analyst 2010;135:230–67. https://doi.org/10.1039/B918972F.

[64] Vapnik V. The Nature of Statistical Learning Theory. Springer New York; 1999.

[65] Burges CJC. A Tutorial on Support Vector Machines for Pattern Recognition. Data Min Knowl Discov 1998;2:121–67. https://doi.org/10.1023/A:1009715923555.

[66] Cristianini N, Shawe-Taylor J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge: Cambridge University Press; 2000. https://doi.org/DOI: 10.1017/CBO9780511801389.

[67] Lam HK, Ling SH, Ling SSH, Nguyen HT. Computational Intelligence and Its Applications: Evolutionary Computation, Fuzzy Logic, Neural Network and Support Vector Machine Techniques. Imperial College Press; 2012.

[68] Smola AJ, Schölkopf B. A tutorial on support vector regression. Stat Comput 2004;14:199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88.

[69] Fix E, Hodges JL. Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties. USAF School of Aviation Medicine; 1951.

[70] Liu T, Moore AW, Gray A, Yang K. An Investigation of Practical Approximate Nearest Neighbor Algorithms. Proc. 17th Int. Conf. Neural Inf. Process. Syst., Cambridge, MA, USA: MIT Press; 2004, p. 825–832.

[71] Selim GEI, Hemdan EE-D, Shehata AM, El-Fishawy NA. Anomaly events classification and detection system in critical industrial internet of things infrastructure using machine learning algorithms. Multimed Tools Appl 2021;80:12619–40. https://doi.org/10.1007/s11042-020-10354-1.

[72] Alatoom YI, Al-Hamdan AB. Signal_delay_data_Jordan 2024.

[73] Haldan M. How Much Training Data Do You Need? 2015.

[74] Al-Omari B, Musa Y. ESTIMATION OF BASE SATURATION FLOW RATES FOR SIGNALIZED INTERSECTIONS IN JORDAN AND KUWAIT. J Eng Appl Sci 2020;15:2995–3000.

[75] Demšar J, Curk T, Erjavec A, Gorup Č, Hočevar T, Milutinovič M, et al. Orange: Data Mining Toolbox in Python. J Mach Learn Res 2013;14:2349–53.

[76] Liping F, Bruce H. Delay variability at signalized intersection. Report No. 1710. Washington, D.C.: Transportation Research Record; 1999.

[77] Liu FT, Ting KM, Zhou Z-H. Isolation Forest. 2008 Eighth IEEE Int. Conf. Data Min., 2008, p. 413–22. https://doi.org/10.1109/ICDM.2008.17.

[78] PySynth. PySynth: Dataset Synthesis for Python 2019.